

Machine Learning and Data Science Application for Financial Price Prediction and Portfolio Optimization

by

Joy Dip Das

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
August 2024

© Copyright 2024 by Joy Dip Das

Machine Learning and Data Science Application for Financial Price Prediction and Portfolio Optimization

Abstract

This thesis explores interconnected advanced machine learning (ML) and data science (DS) methodologies for improved predictive accuracy in financial markets and resilient portfolio optimization through a series of studies.

Studying the literature on ML/DS methodologies extensively led us to observe a significant lack of application of these advances, such as autoencoder (AE), recurrent neural networks (RNN), *etc.* in the finance industry. The novelty of this thesis is to study the price prediction and portfolio optimization applications with the RNN and AE algorithms. Furthermore, unsupervised ML strategies were studied to introduce robustness in the portfolio optimization methods. For this purpose, two innovative encoder-decoder-based RNN architectures autoencoder-based gated recurrent unit (AE-GRU) and autoencoder-based long short-term memory (AE-LSTM) were proposed, which were shown to be effective in predictive efficacy across diverse asset types and market conditions, showcasing enhanced predictive accuracy for financial assets. Various DS concepts, such as statistical data exploration, data visualization, Bollinger bands, data-driven volatility estimates, unsupervised ML *etc.* were integrated while implementing and experimenting with new architectures for price prediction and portfolio optimization.

The proposed models in this thesis showed effectiveness in price prediction and portfolio optimization under varying market conditions. The study also highlights the benefits of diversified portfolios by proposing a novel DL-based model for portfolio construction, especially when coupled with affinity propagation (AP) clustering and appropriate data-driven risk measures based on volatility estimates - with sign correlation (VES) and volatility correlation (VEV).

Traditional models like mean-variance optimization (MVO), Sharpe ratio optimization (SRO), and constrained portfolio optimization (CPO) optimize portfolio weights using objective functions, while recent innovations emphasize data-driven risk measures for minimum risk weights from random samples. Despite challenges with short-term data featuring negative mean returns, the proposed ML-based diversification approaches (for both traditional and data-driven PO) identified portfolios with positive returns by clustering assets with high mean returns.

In summary, this thesis introduces novel hybridized ML/DL models for price prediction and approaches to enhance diversification within specific types of portfolio optimization through advanced clustering methods like affinity propagation and DBSCAN. Diversified stock selection with clustering techniques significantly enhances profitability in data-driven portfolio optimization. These studies underscore its critical role in portfolio resilience, showing that optimal asset selection across different clusters is pivotal for portfolio performance under challenging market conditions.

Contents

Abstract	ii
Table of Contents	vi
List of Figures	vii
List of Tables	ix
Acknowledgments	x
Dedication	1
1 Introduction	1
1.1 Financial Market: Stock and Cryptocurrency	1
1.2 Traditional Techniques in Price Prediction and Portfolio Optimization	3
1.3 Machine Learning and Deep Learning	5
1.4 Objective	8
2 Literature Review	9
2.1 Algorithmic Strategies in Financial Market	9
2.1.1 Application of ML	10
2.1.2 Application of DL	11
2.1.3 Application of Recurrent Neural Network (RNN)	12
2.1.4 Application of hybridized models in financial market	13
2.2 Motivation	14
3 Machine Learning Approaches in Financial Predictions and Portfolio Optimization	17
3.1 Machine Learning Approaches in Price Prediction	17
3.1.1 Bidirectional Long Short-Term Memory (BiLSTM)	18
3.1.2 Bidirectional Gated Recurrent Unit (BiGRU)	19
3.1.3 Activation functions	22
A. ReLU	22
B. ELU	23
C. Tanh	23
3.1.4 Autoencoder	24
3.2 Portfolio Optimization Techniques	25
3.2.1 Traditional Portfolio Optimization	25
A. Equal Weighted Portfolio Optimization (EW)	25
B. Inverse Volatility Weighted Portfolio Optimization (IVW)	26
C. Mean-Variance Optimization (MVO)	27
D. Constrained Portfolio Optimization (CPO)	27
E. Sharpe Ratio Optimization (SRO)	28
3.2.2 Data-Driven Portfolio Optimization	29
A. Risk Measures	29

3.2.3	Diversification	30
3.2.4	Clustering Approaches	30
3.3	Data Science Concepts	32
3.3.1	Data Pre-Processing	32
	A. Data Inference	33
	B. Standardization and Normalization	33
	C. Handling missing Data	34
3.3.2	Statistical Data Exploration	36
	A. Financial Data Distribution	36
3.3.3	Bollinger Bands	36
3.3.4	Unsupervised Learning and Clustering Techniques	38
4	Experimentation	39
4.1	Encoder-Decoder Architecture for Financial Asset Price Prediction	39
4.1.1	Dataset	40
4.1.2	Exploratory data analysis	41
4.1.3	Implementation	42
	A. Stacked AE-LSTM and AE-GRU architecture	42
	B. Hyperparameter Tuning	48
	C. Evaluation Metrics	52
4.2	Hybrid Data-Driven DL-based PO	52
4.2.1	Dataset	53
4.2.2	Implementation	54
	A. Exploratory Analysis	54
	B. Model Design	55
	C. Evaluation Metrics	56
4.2.3	Portfolio Construction and Optimization	57
4.3	Resiliency in PO with Clustering technique	58
4.3.1	Dataset	58
4.3.2	Stock and Cryptocurrency Selection through Clustering	60
	A. Clustering for stock selection	60
	B. Clustering for cryptocurrency selection	61
4.3.3	Portfolio Construction and Optimization	61
4.4	Investigation of Clustering Algorithms for Diversification	62
4.4.1	Dataset	63
4.4.2	Implementation	63
	A. Exploratory analysis	63
	B. Clustering for stock selection	64
4.4.3	Portfolio Construction and Optimization	65
5	Results, Evaluation, and Discussion	66
5.1	Encoder-Decoder Architecture for Financial Price Prediction	66
5.1.1	Predictions for Individual Stock	66
5.1.2	Predictions for the Stock Index- S&P	69
5.1.3	Predictions for the Cryptocurrency- Bitcoin	70
5.1.4	Assessing the Efficacy against the Baseline LSTM and GRU Models	71
5.1.5	Practical Applications	72
5.1.6	Key Findings	73
5.2	Hybrid Data-Driven DL-based PO	74
5.2.1	Performance Analysis of BiLSTM and BiGRU	74

5.2.2	Portfolio Performance	77
	Without Diversification	78
	With Diversification through AP	78
5.2.3	Key Findings	82
5.3	Resiliency in PO with Clustering Technique	83
5.3.1	Stock-based PO	83
5.3.2	Cryptocurrency-based PO	85
5.3.3	Key Findings	85
5.4	Investigation of Clustering Algorithms for Diversification	87
5.4.1	Clustering on EW and IVW	87
5.4.2	Clustering with MVO and CPO	88
	A. Constrained Portfolio Optimization (CPO)	89
5.4.3	Clustering with SRO	91
5.4.4	Clustering with Data-Driven(DD) Approach	91
	A. Data-Driven PO Without Clustering	91
	B. Data-Driven PO With Clustering	92
5.4.5	Key Findings	93
6	Conclusion, Future Works, and Contributions	95
6.1	Conclusions	95
6.2	Future Works	97
6.3	List of Published Contributions	98
A	Algorithms	100
A.1	LSTM	100
A.2	GRU	101
	Bibliography	113

List of Figures

3.1	A basic LSTM chain architecture	18
3.2	A BiLSTM architecture	19
3.3	A GRU architecture	20
3.4	BiGRU architecture	21
3.5	Activation functions.	22
3.6	Autoencoder architecture	24
3.7	20-day Bollinger Bands for 2022	37
4.1	Bollinger plots for financial assets from 2022 to 2023.	42
4.2	Change in volatility.	43
4.3	Encoder-decoder-based novel LSTM and GRU architecture.	45
4.4	Bollinger bands for stocks and cryptos from 2019 to 2023	54
4.5	Proposed Algorithmic Architecture for Portfolio Optimization	55
5.1	Trend prediction by AE-GRU and AE-LSTM for individual stocks.	67
5.2	Trend prediction by AE-GRU and AE-LSTM for stock index.	70
5.3	Trend prediction by AE-GRU and AE-LSTM for cryptocurrency.	70
5.4	Trend prediction by LSTM and GRU for AAPL stock.	71
5.5	Correlation analysis using actual data and forecasted data of 2023	74
5.6	BiLSTM and BiGRU predictions	77
5.7	Performance of portfolio based on data-driven risk measures	79
5.8	Portfolio Weight Distribution based on risk and Sharpe ratio: actual and pre- dicted data	80
5.9	Performance of portfolio based on data-driven risk measures	81
5.10	Portfolio Weight Distribution based on risk and Sharpe ratio with actual and predicted data	82
5.11	Efficient Frontiers using mean-variance, constraint portfolio optimization and data-driven approach (2022)	84
5.12	Efficient Frontiers for technological stock-based portfolios using mean-variance optimization (left) and data-driven approach (right) for 2020	85
5.13	Efficient Frontiers for cryptocurrency-based portfolios using mean-variance opti- mization and data-driven approach	86
5.14	Efficient Frontiers with crypto index using constrained portfolio optimization . .	86
5.15	MVO with different clustering algorithms for 2020	89
5.16	MVO with kmeans clustering algorithm for 2022	89
5.17	CPO with kmeans clustering algorithm(left) and AP(right) in 2020	90
5.18	CPO with clustering algorithms for 2022	90
5.19	Data-driven portfolio optimization without clustering	92

5.20	Data-driven approach with Hierarchical (left) and AP(right) clustering algorithms for 2020	92
5.21	Data-driven approach with different clustering algorithms for 2022	93
A.1	LSTM architecture	101
A.2	GRU architecture	102

List of Tables

4.1	Average Volatility based on 21 rolling days.	41
4.2	Activation functions for different financial assets in AE-LSTM and AE-GRU models.	49
4.3	Volatility with 20 days rolling window	55
4.4	Stocks and Cryptocurrencies	59
4.5	Summary Statistics - Stocks and Cryptocurrencies	59
4.6	Clusters with mean returns	60
4.7	Beta Values of Cryptocurrencies	61
4.8	Crypto-based Portfolio Clusters	61
4.9	Number of clusters using different clustering techniques	64
5.1	Evaluation of the performance of AE-LSTM and AE-GRU through metrics. . . .	68
5.2	Evaluation Metrics for AAPL using LSTM and GRU models	72
5.3	Performance Comparison of BiLSTM and BiGRU	76
5.4	Summary Results for Equal Weight Portfolio	84
5.5	Summary Results for Inversly Volatility Weighted Portfolio	84
5.6	Performance using different clustering algorithms for E.W and I.V.W constructions	88
5.7	SRO with different clustering techniques	91
5.8	Comparison between Traditional and Data-driven PO Methods	94

Acknowledgments

I want to begin by thanking my spiritual guru Shri Anukul Thakur, my advisor Dr. Ruppak. Thulasiram, my committee, my parents, and all the people who have supported me along the way.

Chapter 1

Introduction

This chapter provides an overview of the financial market and its components, discusses traditional and advanced investment strategies, and outlines the objectives of this thesis.

1.1 Financial Market: Stock and Cryptocurrency

The stock market allows individuals to buy and sell shares or ownership stakes in publicly traded companies. Moreover, it enables businesses to raise funds for expansions, research, and development by selling shares to interested investors. This market provides an avenue for individuals and businesses to potentially grow their wealth and plan for their financial future through smart investments. Market dynamics are influenced by diverse elements, including economic indicators, company performance, geopolitical events, and public sentiment. In simple terms, the stock market is like a giant puzzle with many pieces. Various stock products encompass a wide array of financial instruments available in the market. These products can include stocks of individual companies, exchange-traded funds (ETFs) that bundle multiple stocks, mutual funds comprising a diversified portfolio of stocks managed by professionals, and index funds tracking specific market indices. Additionally, there are options and futures contracts based on stocks, providing alternative investment and hedging opportunities. Each of these products has distinct characteristics and risks, catering to different investor preferences and strategies.

Digital assets such as cryptocurrencies are also gaining popularity nowadays opening a new avenue in the realm of finance. Cryptocurrencies are digital or virtual forms of currency that use cryptography for security. They operate on a decentralized network called blockchain, which records all transactions securely and transparently. This technology ensures the integrity and traceability of transactions without the need for a central authority, like a bank or government. Some popular cryptocurrencies include Bitcoin, Ethereum, and Ripple. Bitcoin, the first and most well-known cryptocurrency, focuses on peer-to-peer transactions and decentralized financial systems. Ethereum allows the creation of smart contracts and decentralized applications, broadening its use cases. Ripple, on the other hand, targets cross-border payments and transactions between financial institutions, aiming for faster and more cost-effective transfers. Each cryptocurrency has its unique features and intended applications in the digital economy. Each one works a bit differently and has its own job in the digital money world. In the realm of cryptocurrency trading, various products cater to traders' needs. These encompass trading platforms, wallets, and analytical tools. Trading platforms facilitate buying, selling, and exchanging cryptocurrencies, offering features like order execution and market analysis. Wallets, on the other hand, provide a secure space to store digital assets. Additionally, analytical tools assist traders in making informed decisions by offering insights into market trends and price movements. Overall, the cryptocurrency trading landscape is augmented by these essential products. Cryptocurrencies are highly volatile. The dynamic nature of cryptocurrency stems from its constantly changing values, market conditions, and technological advancements. Cryptocurrencies experience volatile price fluctuations influenced by factors like investor sentiment, regulatory developments, and overall market demand. Additionally, technological innovations and upgrades in blockchain technology can impact how cryptocurrencies operate and are perceived. These shifts and changes make the cryptocurrency landscape highly dynamic, requiring a proactive approach for effective engagement and adaptation.

Investments in financial assets are achieved through portfolio construction as well. A portfolio is a collection of financial assets held by an individual or institution. These assets can include stocks, cryptocurrencies, bonds, mutual funds, real estate, commodities, and other in-

vestments. The purpose of a portfolio is to achieve a balanced and diversified mix of investments that aligns with the investor's financial goals, risk tolerance, and investment horizon. Portfolio optimization (PO) is a financial strategy that aims to select the best mix of investments to achieve a specific objective, such as maximizing return or minimizing risk. This process involves allocating resources among different financial assets like stocks, bonds, and other securities to construct an investment portfolio that aligns with an investor's risk tolerance, investment goals, and time horizon.

The success of investments is heavily dependent on managing risks. The global financial crisis of 2008 necessitated financial markets to develop sustainable forecasting models that consider market risks [1]. Financial institutions are now required to use more advanced credit scoring models to enhance capital allocation and to improve risk management, according to the Basel Committee on Banking Supervision's Basel III monitoring report [2] in 2021. Resilient portfolio construction by involving the possible risk factors is very important in averting drastic market events. Diversity is an important factor for consideration while constructing a risk-averse portfolio. Choueifaty and Coignard [3] put forward the concept of diversification in PO. They introduced a term called 'diversification ratio' for measuring the diversity in a portfolio.

In essence, the financial market consists of various asset types with intricate data distributions. The task of selecting appropriate financial assets, forecasting future trends, and diversifying investment strategies is challenging due to these complexities. This research aims to address these challenges and propose effective strategies to tackle them.

1.2 Traditional Techniques in Price Prediction and Portfolio Optimization

Various financial instruments and cryptocurrencies introduce complexities due to their volatile nature for buying and selling ownership of shares. The dynamics of the stock and cryptocurrency markets are influenced by a multitude of factors such as political, geographical, and socio-economic considerations. The pronounced variability across these diverse factors

contributes to fluctuations in stock market trends.

Predicting stock prices for informed investment decisions involves traditional methods like analyzing historical data and patterns. These commonly involve examining past stock price data through statistical techniques like the use of moving averages (see [4]), autoregressive integrated moving average (ARIMA) modeling (see [5]), exponential smoothing (see [6]), and so forth. These techniques often overlook the temporal dependencies present in stock data, a crucial factor for accurate stock market prediction.

The most traditional technique for portfolio optimization (PO) is mean-variance optimization (MVO), an idea proposed by Harry Markowitz [7] in 1952. Its objective is to either maximize anticipated returns for a specified risk level or minimize risk for a targeted return. The resultant efficient frontier illustrates optimal trade-offs between risk and return. Expanding upon the foundations laid by Markowitz, modern portfolio theory (MPT)[8] underscores the importance of diversification in attaining an ideal equilibrium between risk and return, taking into account the interrelation among diverse assets. Another traditional technique, the capital asset pricing model (CAPM) incorporates the risk-free rate, market risk premium, and the beta¹ associated with the individual asset for computing the expected yield of an asset [9; 10; 11; 12]. This facilitates the construction of portfolios by assessing the equilibrium between risk and anticipated returns. The CAPM revolutionized asset pricing theory; although widely taught and applied, its simplicity and intuitive appeal are marred by empirical challenges, notably poor performance in real-world applications, possibly stemming from inherent model limitations or inadequacies in market proxies used for testing and application. The arbitrage pricing theory (APT) [13] extends the horizon of PO by incorporating diverse factors influencing asset valuations. This allows for a thorough exploration of the interrelationship between risk and return within financial contexts. The Black-Litterman model [14], in the realm of dynamic management of portfolio emphasizing information utilization, operates within a

¹The beta measure of volatility, commonly referred to as beta (β), quantifies the sensitivity of an asset's returns relative to the returns of a benchmark index. Specifically, it indicates how much an asset's price is expected to move in response to changes in the overall market. A beta of 1 suggests that the asset's price will move in tandem with the market, a beta greater than 1 indicates higher volatility than the market, and a beta less than 1 implies lower volatility. Negative beta values suggest an inverse relationship with the market.

Bayesian analytic framework, allowing the portfolio manager to express views, subsequently translated into security return forecasts, offering a theoretically and practically appealing tool for portfolio construction despite persisting challenges.

The conventional method of estimating asset weights includes creating portfolios with equal weights and inverse volatility weights [15]. Various alternative approaches, such as the enhancement of the Sharpe ratio optimization (SRO), optimization of mean-variance (MVO), and portfolio construction through user constraints (CPO) have been explored in this context [15]. Sharpe [16] introduced a mathematical measure to assess portfolio performance, providing a framework for evaluating different PO techniques.

These traditional PO methods explained above have a myriad of weaknesses in practical scenarios. One significant limitation of MVO and MPT lies in their sensitivity to input data, as reliance on historical information for expected returns and risk parameters can lead to suboptimal portfolios when market conditions deviate from historical patterns. Additionally, the assumption of normality in asset returns and the focus on mean and variance may not accurately capture the non-normal and complex distributions observed in real financial markets, leading to incorrect estimates of risk. These methods often underestimate tail risks², neglecting the potential impact of extreme events on portfolio performance. Furthermore, their single-period analysis overlooks the dynamic nature of financial markets, and the static approach may not adapt well to changing economic environments over time.

1.3 Machine Learning and Deep Learning

Various studies indicate that machine learning (ML) and deep learning (DL)³ methodologies exhibit superior performance in stock price forecasting when contrasted with conventional statistical methods. This superiority arises from their adeptness in managing intricate, non-linear

²Tail risks are the risks associated with rare and extreme events that lie at the far ends of a distribution of investment returns. These events, although infrequent, can have a substantial impact on portfolio performance and can lead to significant losses.

³Deep learning is a subset of the field of machine learning. In this thesis, unless stated otherwise, we will use the term machine learning to refer to traditional (such as shallow or non-deep learning alternatives) machine learning methods, and deep learning to refer to more recent models with many layers, large number of parameters, or both.

patterns and handling extensive datasets [17],[18].

ML algorithms are now being heavily explored in stock price prediction [19]. Traditional ML-based regression techniques such as linear regression (see [20]), support vector regression (SVR) [21], decision tree regression [22], random forest (RF) regression [23], *etc.* are used in stock prediction. Moreover, these traditional techniques have also been pursued in cryptocurrency prediction [24]. DL algorithms such as artificial neural networks (ANN), convolutional neural networks (CNN), recurrent neural networks (RNN), *etc.* are also used in stock market prediction [25]. ANN and CNN do not consider temporal dependencies, and hence RNN algorithms, for example, long short-term memory (LSTM) and gated recurrent unit (GRU) are heavily used in stock market prediction [26]. These DL techniques are also used for highly volatile cryptocurrency price prediction [27]. They are good at finding subtle patterns and trends in financial data that other methods might miss. These RNN algorithms comprise gating mechanisms that regulate information flow within the network, allowing for better capturing of long-term dependencies in the input data. LSTM, addressing vanishing and exploding gradient issues in traditional RNNs, is designed to model intricate patterns in sequential data efficiently [28]. Similarly, GRU is also a popular RNN algorithm used in stock price prediction. A GRU is a type of RNN architecture designed to model sequential data. In terms of computational speed, the GRU typically demonstrates smaller training times than LSTM [29]. This can be attributed to GRU's simpler architecture, involving only two gates compared to LSTM's three gates. The reduced complexity leads to fewer parameters to compute and consequently faster convergence during training.

ML and DL techniques are also heavily applied in PO. ML-based PO is garnering attention because of its potential for capturing intricate, non-linear information and vast datasets in financial markets, outperforming traditional methods. Designed to tackle challenges in PO related to both mean-variance and mean conditional value at risk (CVaR), ML-based performance-based regularization (PBR) has been developed [30]. Other than that, ML techniques are fused with MVO for designing portfolio investment strategies [31; 32]. DL models are utilized to directly optimize the Sharpe ratio [33]. In addition to the models mentioned above, the autoencoder is

also used as a popular DL model in computational linguistics and representation learning [34]. Autoencoders (discussed later in this thesis) are also used for constructing asset pricing models [35].

In summary, ML and DL algorithms are being relentlessly researched in various fields. computational finance (CF) and financial technology (FinTech) research areas are adapting these algorithms and designing trading technologies. Clustering, classification, prediction, *etc.*, rely on the ML and DL algorithms.

Nowadays, ML and DL algorithms are being hybridized in order to achieve different goals set by market pundits. Recent research on applying these hybridized models in the finance market focuses on smart profitable PO and financial asset management. From stock prediction to PO, ML algorithms are being explored rigorously to find useful models in varying risk scenarios.

Hybridization of ML and DL algorithms with each other and with other statistical methods can result in powerful algorithmic strategies in CF. These algorithms can be effectively utilized in enhancing predictive accuracies and developing profitable PO techniques. Hybridization of established or traditional ML methods with newer DL architectures broadens the spectrum of applications and can improve performance across several domains. LSTM-GRU hybridization is being heavily explored in the financial time series forecasting problem domains. Besides that, LSTM-CNN and GRU-CNN also have been explored for different problem domains including finance [36]. Other than these models, LSTM-GRU-ARIMA [37] is also a popular hybridized model that can be used in different problem domains. Rubell et al. [38] conducted experiments employing a multivariate sequential LSTM autoencoder on various stocks, including AAPL, GOOGLE, JPM, JNJ, *etc.*, demonstrating its superiority over univariate sequential LSTM autoencoder, GRU, and generative adversarial networks (GAN).

Recent advances in hybridized ML and DL models have led researchers to explore the efficiency of various hybridized DL models for PO [39]. The integration of ML and DL models could potentially lead to a prominent advancement across diverse domains.

1.4 Objective

Financial forecasting, diversification in PO, and selecting financial assets amid complex data distributions and unpredictable crisis periods present significant challenges. This research aims to address these challenges by designing effective models and diversifying asset selection for risk-averse investments. Specifically, in this research, we propose a hybridized approach and do thorough experimentation that enhances predictive accuracy for forecasting trends of various financial assets under different market conditions. Additionally, a hybrid DL-based approach for efficient PO is proposed. To further enhance portfolio construction techniques and increase the resiliency of existing methods, clustering algorithms are experimented with to introduce diversification.

The remainder of this thesis is structured as follows: Chapter 2 reviews the related works on using ML and DL techniques in financial asset forecasting and PO, discussing the motivation behind this study. Chapter 3 discusses the models and methodologies. Chapter 4 details the experimental setup and implementation of the models. Chapter 5 presents the results obtained from the experiments and their evaluations. Finally, Chapter 6 concludes the thesis and discusses the published contributions from this research.

Chapter 2

Literature Review

Stock market data is notably intricate, characterized by complex interconnections among its components. The dynamics of the stock market are influenced by a multitude of factors such as political, geographical, and socio-economic considerations. The pronounced variability across these diverse factors contributes to fluctuations in stock market trends. Particularly during critical events, accurately predicting stock market performance becomes notably challenging and of utmost importance. Different researchers have used ML and DL algorithms to explore stock market prediction and PO. ML techniques are being heavily explored in the arena of finance and these algorithms are superior to traditional techniques of forecasting such as ARIMA, as the traditional approaches cannot identify internal dependencies of the data [40].

This chapter discusses many important ML and DL techniques applied to finance problems that naturally lead to the motivation of this current study.

2.1 Algorithmic Strategies in Financial Market

In this section, different algorithmic procedures for financial forecasting and PO are discussed. This section is further subdivided into several subsections to discuss the published literature on ML, DL, RNN, and hybridized approaches in the financial market.

2.1.1 Application of ML

ML has grown tremendously in many areas of research and applications. It would be a daunting task to comprehensively cover all work from literature from multiple areas and hence our focus in this section is limited to financial applications.

Dimensionality reduction techniques can reduce the redundancy in the stock market data and can result in efficient price prediction. Zhong and Enke [41] have used dimensionality reduction techniques along with ANN for daily stock market return. However, their dimensionality reduction techniques utilized different variations of principle component analysis (PCA) which only capture linear/planar intricacies among the data. The stock market is mainly affected by non-linear complexities. Kohli et al. [42] forecasted the movements of the Bombay Stock Exchange (BSE) by considering various factors such as market history, commodity prices, and foreign exchange rates. They have found gold prices to affect BSE the most. Their analysis revealed the superiority of the AdaBoost algorithm among others explored. Awoye [43] investigated the use of ML techniques to enhance portfolio returns by applying the minimum variance model of Markowitz. Du [44] has employed a comparable methodology, utilizing ML algorithms like support vector machine (SVM) and LSTM for stock price prediction, followed by the application of MVO to construct portfolios. Tola et al. [45] utilized cluster analysis for PO, showcasing its ability to enhance the predicted portfolio's reliability. With current advances in ML techniques, Behera et al. [46] recommended an ML-based technique for selecting assets, which was followed by mean-VaR portfolio optimization. The study found the approach of the mean-VaR model with AdaBoost prediction to be more efficient than other models that were tested.

Ibanez [47] proposed a portfolio construction technique that emphasizes diversification. This approach employs singular value decomposition to identify underlying factors and utilizes hierarchical agglomerative clustering to address implementation challenges associated with the method. Clustering algorithms can be used in diversifying financial assets and constructing risk-averse portfolios. Li [48] applied k-means clustering to enhance the performance of the global minimum variance (GMV) portfolio. The approach also incorporates regularization within the

autoregressive shrinkage model. Aslam et al. [49] have incorporated a three-factor analysis to cluster stocks and identified clusters comprising the best-performing stocks for portfolio construction. Pranata et al. [50] employed clustering techniques such as the k-means and DBSCAN clustering algorithms to cluster the IDX (Indonesia Stock Exchange) companies. These clustering techniques are commonly employed in stock forecasting and recommendation systems [51] as well as to optimize portfolio construction and risk management [52]. Wang and Aste [53] proposed a PO approach based on inverse covariance clustering, which involves identifying market states and incorporating them into a dynamic PO process.

2.1.2 Application of DL

Application of DL models is rising phenomenally in the arena of stock prediction [54]. Jiang [55] has summarized the recent progress in stock market prediction using different DL models and provided a general workflow for the stock prediction domain using DL. Nikou et al. [56] have considered the non-linearity and non-stationarity of stock market data when they analyzed different ML and DL algorithms in predicting the daily close price data. Their findings suggest DL models surpass traditional ML algorithms. They also noted that SVR and RF perform well next to the DL models. Nabipour et al. [57] have studied nine prominent ML models and two powerful DL models in stock prediction and binary classification. Their research similarly suggests the supremacy of deep learning models. Jang and Seong [58] introduced a PO method that combines modern portfolio theory with DL based on deep reinforcement learning (DRL).

DL algorithms are generally seen to be outperforming ML algorithms due to having neuron-based learning capabilities. Fernández and Góme [59] have explored a particular network, the Hopfield network in PO. Uysal et al. [60] used a risk budgeting model as a layer in the deep neural network which overperformed the traditional techniques. Yu et al. [61] used radial basis function (RBF) neural network and have observed sufficient efficiency. Jang and Seong [58] explored the combination of DRL and MPT and observed substantial improvement in profit maximization. Yang [62] further explored DRL in PO and reported the superior performance of representation learning-based DRL in PO. Ngo et al. [63] did a comparative study on

the performance of DRL with other DL and traditional PO techniques on the emerging and developed stock markets. Their study confirms the better responsiveness of DRL to the market dynamics compared to other algorithms.

2.1.3 Application of Recurrent Neural Network (RNN)

The above-mentioned ML and DL algorithms do not consider temporal dependencies and hence RNN variants such as LSTM and GRU are being explored in stock market behavior prediction, which considers past temporal values of the stock market. Ma et al. [64] suggested the application of Autoencoder and LSTM in PO, utilizing these DL methods to enhance the worst-case omega model. LSTM overperforms ML algorithms like SVR in several markets [65]. However, the architecture of LSTM can be specific market-dependent. Jiang et al. [66] have explored the functionality of LSTM in different markets such as Chinese, European, and American markets. Their findings suggest the superiority of LSTM in the European and American markets compared to the Chinese market. Furthermore, LSTM is found to be inefficient when working with smaller datasets, which is often considered a drawback by certain researchers such as by Sheth and Shah [67]. Apart from LSTM, GRU has also been explored in stock prediction quite extensively [68]. Gupta et al. [69] have performed stock prediction based on LSTM and GRU along with incorporating PCA and least absolute shrinkage and selection operator (LASSO) techniques. Their research also suggests a similarity in the functionalities between LSTM and GRU in the case of prediction. However, the training time of GRU is much smaller than the LSTM because of one fewer gate. Seabe et al. [70] experimented with LSTM, GRU and Bi-LSTM and found the superiority of Bi-LSTM on cryptocurrency prediction. Kim et al. [71] did a comparative study on the performance of LSTM and GRU on cryptocurrency price prediction, Their study shows that GRU performs better than LSTM in certain scenarios. Phan-Thi et al. [72] demonstrated the effectiveness of the bidirectional GRU model combined with the attention mechanism, yielding favorable results for longer-term PO.

Retaining long-term context in financial time series data is very important since predicting future values of financial assets depends heavily on important past events. Hence, the context

of those events should be remembered by the model including important non-linear features of time-series data.

2.1.4 Application of hybridized models in financial market

Hybridization of ML and DL algorithms in price prediction and PO involves combining the strengths of the considered approaches to improve predictive accuracy and investment performance. ML algorithms, known for their efficiency in handling structured data and traditional statistical techniques, are integrated with DL models, which excel in capturing complex patterns and dependencies in large datasets. This synergy enhances the robustness of price-prediction models and optimizes asset allocation strategies in PO, leading to better risk management and higher returns. By leveraging both ML and DL, investors can develop more adaptive and resilient investment strategies in dynamic market conditions.

Popular RNN models such as the LSTM model have been also hybridized in the prediction of the non-stationary and non-linear stock market [73]. Fang et al. [74] have proposed a hybrid forecasting framework using LSTM, batch normalization layer, dropout layer, and binary classifier for trend forecasting of financial time series. Their study used the S&P500, Shanghai Stock Exchange 180, and China Securities Index 300 for experimentation. From their experimentation, it is evident that the performance of hybridized DL models is generalized to different datasets.

For proper prediction, consideration of multiple features is also very important thus hybridization helps in this scenario. Through hybridizing several ML/DL models, multiple features of different algorithms can be incorporated into a single model. It helps to capture previously unnoticed complexities behind the scenes. LSTM has been combined with adaptive boosting (AdaBoost) and k-nearest neighbor (KNN) [75]. Compared to traditional ML models, the performance of this combined approach was significant. Kwak and Lim [76] experimented on the AdaBoost and GRU ensemble model and noticed increased efficiency than the LSTM, GRU, and ARIMA models on the Korean (KOSPII) market. Previously, Shen et al. [77] have experimented with GRU by replacing the last layer with SVM for trading signal

prediction. Their study shows significant performance enhancement when ML and DL models are hybridized. Song and Choi [78] have explored various hybridizations of DL models and observed a significant rise in efficiency compared to other traditional models. They have mainly hybridized LSTM and GRU with CNN to observe the efficiency of the hybridized models in forecasting the financial market. John and Latha [79] have used BiLSTM on GRU and stacked LSTM units to develop a hybrid RNN model and observed enhancement in the efficiency of the hybrid model in stock prediction. Fang et al. [74] have proposed a hybrid forecasting framework using LSTM, batch normalization layer, dropout layer, and binary classifier to forecast the movement of financial time series. Their study used the S&P500, China Securities Index 300, and Shanghai Stock Exchange 180 for their experimentation. From their experimentation, it is clearly evident that the performance of hybridized DL models is generalized to different datasets. Also, in the realm of cryptocurrency prediction, hybridization is a prominent technique. Petrovic et al. [80] used a hybridized LSTM-GRU model along with swarm intelligence for cryptocurrency prediction.

Hybridized DL algorithms are a prominent field of research in PO and the development of investment strategies. Modified deep belief networks (DBN) and RNNs are heavily utilized in PO [81]. Different models such as CNN and RNN are being fused for stock selection and optimization for the formation of profitable risk-averse portfolios [82]. Moreover, various heuristics are also being explored by the amalgamation of these with various DL algorithms for PO [83].

2.2 Motivation

In computational finance, hybridization primarily aims to integrate a range of ML and DL models across diverse market contexts. However, their efficacy varies across distinct market conditions. Presently, there is an absence of well-established models that comprehensively account for market dynamics and exhibit performance across diverse market scenarios. Additionally, the financial dataset encompasses a multitude of underlying features necessitating meticulous consideration through diverse statistical measures. This research involves design

and experimentation for novel hybridization approaches for retaining long-term context and non-linear important features from past observation. The main objective is to augment and accelerate predictive accuracy, particularly in the context of forecasting stock prices, stock indices, and cryptocurrency values. Retaining long-term context in financial time series data is very important since predicting future values of financial assets depends heavily on important past events. Hence, the context of those events should be remembered by the model including important non-linear features of time-series data.

LSTM and GRU are commonly employed in the prediction of stocks and cryptocurrencies. However, these basic LSTM and GRU structures do not take into account the significance of non-linear features. In natural language processing (NLP), both LSTM and GRU encounter difficulties in predicting words when the sentence length increases significantly. Consequently, these algorithms struggle to remember the context over a very long term. To address this issue, the Transformer Network (TN) was introduced, which uses an attention mechanism in an encoder-decoder architecture to remember the long-term context [84]. TN architectures are computationally resource-heavy and take a long time to process information. For effective financial price prediction and PO, resource-friendly architectures are needed. One of the major concerns of this research is to enhance predictive accuracy using LSTM/GRU units hybridized in an encoder-decoder fashion, to increase the information retention capacity of these RNN algorithms.

Traditional PO techniques do not consider the non-linearity in the underlying financial data. Building upon prior work, Bowala and Singh [85] proposed a data-driven method for composing a portfolio comprising conventional stocks and cryptocurrencies (cryptos). They explored statistical risk measures and put forward a range of statistical correlations for assessing portfolio composition utilizing these metrics. In this research, experimentations have been done with various clustering techniques alongside both traditional and data-driven PO methods to enhance investment strategies by introducing diversification. Such experimenting improved the resiliency of the portfolios and enhanced profitability. Moreover, this research also explored the amalgamation of the DL algorithm with the data-driven approach to propose a novel hybridized

portfolio construction technique.

Chapter 3

Machine Learning Approaches in Financial Predictions and Portfolio Optimization

ML and DL models play a crucial role in financial price prediction research. Market experts employ a variety of these models for PO to achieve better accuracy and resilience in their investment strategies. This chapter provides an in-depth discussion of ML, DL, and DS models used in this thesis, highlighting their application and effectiveness in forecasting financial asset prices and optimizing portfolios. Additionally, it examines traditional (mean-variance optimization, Sharpe ratio optimization, *etc.*) and data-driven approaches for portfolio optimization in current financial market conditions and compares the portfolio optimization with recently proposed risk measures based on sign and volatility correlations.

3.1 Machine Learning Approaches in Price Prediction

Bidirectional RNNs (LSTM and GRU) harness the power of simultaneous forward and backward information flow, enhancing their ability to capture context from both past and future sequences. This dual-directional processing enables more robust modeling of temporal dependencies in sequential data, making them effective for tasks requiring comprehensive context

understanding. The basic LSTM and GRU architectures, though they are well-known and well-studied, are described in the Appendix A for the sake of completeness.

3.1.1 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM is an advancement of the conventional LSTM architecture, specifically crafted to consider dependencies in both forward and backward directions within sequential data. LSTM, a variant of RNN is specifically efficient in considering dependencies of long-term time series. BiLSTM enhances the capabilities of traditional LSTMs by processing input sequences in both forward and backward directions simultaneously.

Figure 3.1 shows a basic chain architecture of LSTM¹. This chain architecture of LSTM represents how sequential data is processed, where the information passes in a single direction.

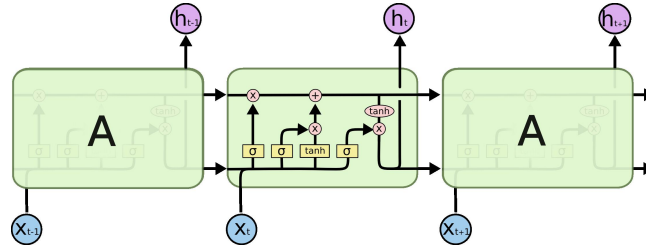


Figure 3.1: A basic LSTM chain architecture

In the Figure 3.1, the forget gate, determines the information to be excluded from the cell state. The input gate decides which values to update in the cell state. The cell state is responsible for storing long-term information. The output gate controls the information to be output from the cell state. h_t represents the hidden state, encompassing short-term information for output.

Figure 3.2 demonstrates a BiLSTM architecture². Within an LSTM unit, key components encompass a cell, an input gate, a forget gate, and an output gate. These components work together to selectively retain and update information over sequential steps. The key innovation in BiLSTM is its bidirectional architecture. Instead of processing the input sequence only from left to right (as in traditional LSTMs), BiLSTM processes it in both directions simultaneously.

¹<https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (Date accessed: January 2024)

²<https://8f430952.rocketcdn.me/wp-content/uploads/2021/07/image-5.jpeg> (Date accessed: January 2024)

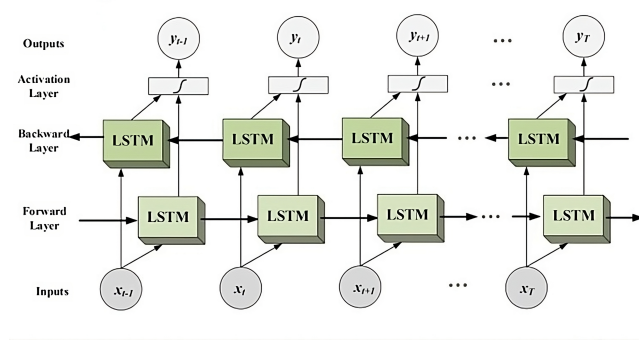


Figure 3.2: A BiLSTM architecture

In the forward pass, the input sequence is handled from the start to the finish. The hidden states at each step capture the information up to that point, considering the context from the past. Simultaneously, in the backward pass, the input sequence is processed from the end to the beginning. The hidden states in this direction capture information considering the context of the future. At each time step, the concatenation of hidden states from both the forward and backward passes takes place. This results in a representation that encodes information from both the past and the future, providing a more comprehensive context for each element in the sequence. By considering bidirectional dependencies, BiLSTM has an enhanced ability to capture long-range dependencies and understand the context in which each element appears within the sequence. BiLSTM is trained using *backpropagation through time* (BPTT)³ and optimized using techniques like *gradient clipping*⁴ to address vanishing or exploding gradient issues. Algorithm 1 captures the steps in the BiLSTM model used in this research. The term backward pass here refers to processing the input sequence from right to left, which differs from backpropagation.

3.1.2 Bidirectional Gated Recurrent Unit (BiGRU)

BiGRU is an advancement of the traditional GRU neural network architecture designed to capture bidirectional dependencies in sequential data. GRU is a variant of RNN, like LSTM,

³Backpropagation through time (BPTT) is an algorithm used to train recurrent neural networks by unfolding them through time and applying backpropagation.

⁴Gradient clipping is a technique used to prevent exploding gradients by capping the gradients during backpropagation to a maximum value.

Algorithm 1: Bidirectional Long Short-Term Memory (BiLSTM)

Input: Input sequence X of length T

Output: Output sequence h

Initialize forward and backward LSTM networks;

Initialize parameters, weights, and biases;

for $t \leftarrow 1$ **to** T **do**

Forward pass through the forward LSTM network: $h_t^f \leftarrow \text{LSTM}_{\text{forward}}(x_t, h_{t-1}^f, c_{t-1}^f)$;

Backward pass through the backward LSTM network:

$h_t^b \leftarrow \text{LSTM}_{\text{backward}}(x_t, h_{t+1}^b, c_{t+1}^b)$;

Concatenate forward and backward hidden states: $h_t \leftarrow [h_t^f, h_t^b]$;

Perform further processing or use the concatenated states for downstream tasks;

return h

that handles sequential time series. BiGRU enhances the capabilities of GRU by processing input sequences in both forward and backward directions simultaneously.

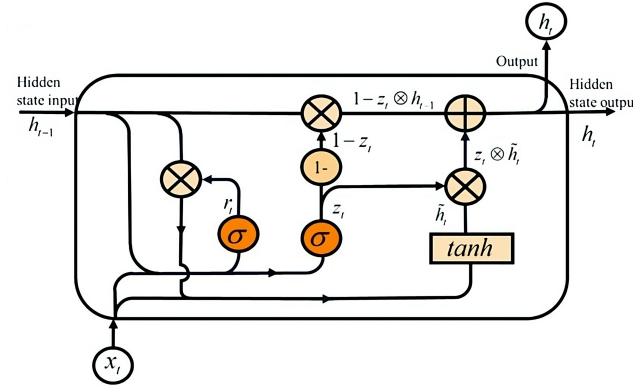


Figure 3.3: A GRU architecture

Figure 3.3 illustrates a basic GRU architecture⁵. It consists of a reset gate and an update gate, allowing it to selectively retain and update information over sequential steps. The reset gate forgets specific past information based on weighted calculation, similarly, the storage of relevant information is decided by weighted calculation by the update gate. This mechanism

⁵https://www.researchgate.net/publication/355705028/figure/fig2/AS:108387081234023101635426347531/GRU-structure-diagram_W640.jpg (Date accessed: January 2024)

helps GRU capture long-term dependencies in sequential data while maintaining computational efficiency. In the GRU model framework as mentioned in the Figure 3.3, r_t acts as the reset gate, and z_t operates as the update gate. The vector $\tilde{h}_t$ signifies a candidate activation vector. The functions σ and $\tanh$ denote the sigmoid and hyperbolic tangent functions, respectively.

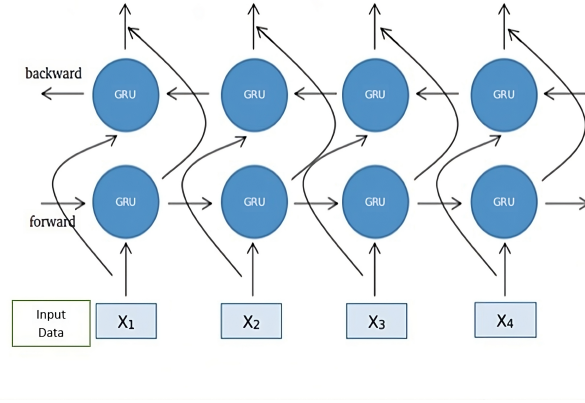


Figure 3.4: BiGRU architecture

Algorithm 2: Bidirectional GRU

Data: Input sequence $\mathbf{X} = (x_1, x_2, \dots, x_T)$

Result: Bidirectional GRU output $\mathbf{H} = (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_T, \overleftarrow{h}_T, \dots, \overleftarrow{h}_1)$

Initialize GRU forward and backward hidden states: $\vec{h}_0 = \overleftarrow{h}_{T+1} = \mathbf{0}$;

for $t = 1$ **to** T **do**

Perform forward GRU step;

$\vec{h}_t = \text{GRU_Step}(\vec{h}_{t-1}, x_t)$;

Perform backward GRU step;

$\overleftarrow{h}_t = \text{GRU_Step}(\overleftarrow{h}_{t+1}, x_t)$;

Concatenate forward and backward hidden states;

$\mathbf{H}[t] = [\vec{h}_t, \overleftarrow{h}_t]$;

return $\mathbf{H}$;

Figure 3.4 illustrates the architecture of BiGRU⁶. BiGRU is employed for tasks involving

⁶<https://www.researchgate.net/publication/366162819/figure/fig1/AS:11431281106354346@1670632077172/BiGRU-model-structure-diagram.jpg> (Date accessed: January 2024)

sequential data analysis, where understanding the context of information requires considering both preceding and succeeding information. The key innovation in BiGRU is its bidirectional architecture. Instead of processing the input sequence only from left to right, as in traditional GRUs, BiGRU processes it in both directions simultaneously similar to the process explained for BiLSTM model earlier. By considering bidirectional dependencies, BiGRU has an enhanced ability to capture long-range dependencies and understand the context in which each element appears within the sequence. Algorithm 2 captures the steps in BiGRU model used in this research. As mentioned earlier, the term backward pass here refers to processing the input sequence from right to left, which differs from backpropagation.

3.1.3 Activation functions

Three popular activation functions are used commonly in the DL models: (1) ReLU, (2) ELU, and (3) tanh. The choice of activation function depends on the dataset and model architectures. Improper choice might result in sub-optimal performance.

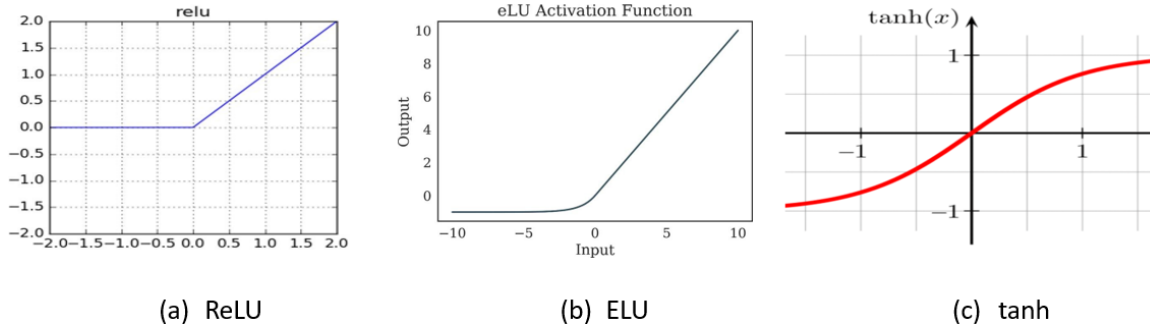


Figure 3.5: Activation functions.

A. ReLU

The Rectified Linear Unit (ReLU) activation function is extensively utilized in neural networks, providing non-linearity by outputting the input directly for positive values and zero for negative ones. Its popularity stems from its simplicity, computational efficiency, and effectiveness in dealing with the vanishing gradient problem during DNN training. ReLU addresses the issue of vanishing gradients that arise due to the saturation of neurons when sigmoid functions

are used. Improper weight initialization can lead to the ‘dying ReLU’ problem causing neurons to die, especially due to the negative saturation.

The formula for ReLU is $f_{Relu}(h_{i,k}) = \max(0, h_{i,k})$.

Figure 3.5(a) presents a graphical representation of ReLU.

B. ELU

The Exponential Linear Unit (ELU) is an activation function commonly used in artificial neural networks. It brings non-linearity to the network by permitting the inclusion of negative values, unlike traditional activation functions such as ReLU. ELU has a smooth curve for negative inputs, avoiding the issue of dead neurons and enabling better convergence during training. It exhibits the ability to capture information from both positive and negative input ranges, contributing to improved learning in DNN. ELU addresses the issue of the dying ReLU problem that occurs due to negative saturation but bad initialization still can cause underperformance. The ELU activation function is defined as:

$$f(k) = \begin{cases} k & \text{if } k > 0 \\ \alpha \cdot (e^k - 1) & \text{if } k \leq 0 \end{cases}.$$

Here, α is a positive hyperparameter controlling the slope of the function for negative inputs.

Figure 3.5(b) presents a graphical representation of ELU.

C. Tanh

The hyperbolic tangent function, denoted as $\tanh(x)$, is a common activation function in neural networks. It squashes input values to the range of $(-1, 1)$, making it zero-centered and aiding in mitigating vanishing gradient problems during training. The tanh activation is particularly useful in scenarios where zero-centered outputs are desired and has applications in various layers of neural network architectures. The tanh activation function delivers output centered around zero, but it can lead to the vanishing gradient problem due to neuron saturation

caused by improper weight initialization. The formula for the tanh activation function:

$$\tanh(k) = \frac{e^k - e^{-k}}{e^k + e^{-k}}.$$

Figure 3.5(c) presents a graphical representation of tanh.

3.1.4 Autoencoder

An autoencoder is like a smart data compressor and decompressor. It is a DL algorithm that learns to shrink down complex information into a compact representation (encoding) and then expand it back to its original form (decoding). This helps in capturing the essential features of the data while reducing unnecessary details, making it easier to analyze and use. Figure 3.6 provides a basic visualization of the autoencoder architecture⁷.

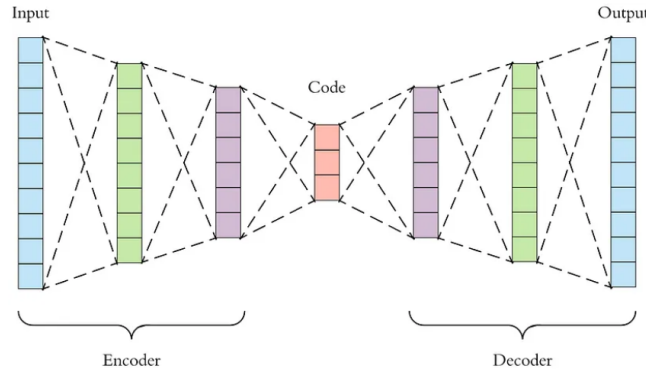


Figure 3.6: Autoencoder architecture

A generalized data flow of an autoencoder experimented in this research is as follows:

$$x \xrightarrow{\text{Encoder}} h = e(x) \xrightarrow{\text{Decoder}} \hat{x} = d(h).$$

Here, an encoder is a function that takes input x and is defined as

$$h = e(x).$$

The architecture of an autoencoder is designed to extract the hidden representation of input data through its encoder component. This representation, which exists in a reduced dimensional

⁷<https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798> (Date accessed: January 2024)

space, allows the model to discern non-linear characteristics of the input feature vector. In forecasting time series, these non-linear characteristics can offer valuable insights into the non-linear relationships among data points, which are crucial for forecasting future values. In the given equation, h signifies the hidden representation of the input data as obtained by the encoder component. The decoder component then reconstructs the data, denoted by $\hat{x}$, by minimizing the reconstruction error. This process of reconstruction eliminates noise or outliers that are not correlated with the dataset, leaving behind only the significant features that are relevant to the context of the dataset.

3.2 Portfolio Optimization Techniques

The construction of profitable portfolios has long been a vibrant area of research. While investment strategies developed from this research have their limitations, identifying these limitations has led to the proposal of new models tailored to current market needs. Given the constantly evolving financial market, this thesis aims to address market dynamics and enhance investment strategy resilience by identifying efficient PO algorithms and advancing research through the modification and extension of existing approaches. This section discusses the traditional and contemporary data-driven approaches, as well as the ML-based clustering techniques proposed and utilized in this thesis to enhance diversification in PO.

3.2.1 Traditional Portfolio Optimization

A. Equal Weighted Portfolio Optimization (EW)

In a portfolio utilizing the equal-weighted portfolio optimization technique, every stock is accorded uniform treatment, and an identical proportion of weight is allocated to each individual stock. Specifically, the weight of each stock is calculated as $W_i = \frac{1}{N}$ (see [15]).

B. Inverse Volatility Weighted Portfolio Optimization (IVW)

The inverse volatility-weighted portfolio assigns weights to assets based on the reciprocal of their volatility. Specifically, the weight of each asset is calculated as (see [15]) $W_i = \frac{1}{\sigma_i} \cdot \frac{1}{\sum_j \frac{1}{\sigma_j}}$, where, σ_i denotes the volatility associated with the i -th asset. We employ methodologies encompassing equal-weighting and inverse volatility-weighting to compute the portfolio weights. Subsequently, the yearly mean returns, annual risk, and diversification ratio are computed.

Portfolio return is defined as follows: Let P be comprising of assets $A_1, A_2, \dots, A_N$ with weights $W_1, W_2, \dots, W_N$ and $R_1, R_2, \dots, R_N$ as the asset returns. The portfolio return R_p is determined by a weights summation of individual asset returns given by, $R_p = W_1 R_1 + W_2 R_2 + \dots + W_N R_N = \sum_{i=1}^N W_i R_i$. If there were T_y trading days in a year (e.g. 251 out of 365 days), the portfolio's annualized return can be calculated as (see [15]) $R_{Ann} = \left(\sum_{i=1}^N W_i R_i \right) T_y$. Portfolio risk is the standard deviation of its returns and is given by (see [15]), $\sigma = \sqrt{\sum_i \sum_j W_i W_j \sigma_{ij}}$.

The variance-covariance matrix of returns, which represents the covariance of returns between assets i and j of the portfolio, is denoted by σ_{ij} . To calculate the covariance matrix for a dataset with n observations and p variables, we can use the following formula:

$$\sigma = \frac{1}{n-1} \mathbf{X}^\top \mathbf{X}. \quad (3.1)$$

Here,

- $\mathbf{X}$ is an $n \times p$ matrix of observations, with each row representing an observation and each column representing a variable.
- $\mathbf{X}^\top$ is the transpose of the matrix $\mathbf{X}$.

Standard deviation, a statistical measure of data dispersion, characterizes how much the data deviates or spreads from its mean. When it comes to portfolios, the standard deviation of their returns, which represents their risk, is a useful metric for measuring the consistency of their returns. The annualized portfolio risk in percentage, σ_{Ann} , can be computed from (see [15]): $\sigma_{Ann} = \sqrt{\left(\sum_i \sum_j W_i W_j \sigma_{ij} \right) T_y}$.

C. Mean-Variance Optimization (MVO)

MVO has two objectives: to increase the expected portfolio return and to decrease the portfolio's risk.

The MVO model is given by (see [15]):

$$\min - \sum_{i=1}^N W_i R_i + \sqrt{\sum_i \sum_j W_i W_j \sigma_{ij}}$$

subject to (3.2)

$$\sum_{i=1}^N W_i = 1, \quad 0 \leq W_i \leq 1$$

where W_i is portfolio weight, R_i is the portfolio expected return, σ_{ij} is the variance-covariance of returns, and N is the total number of assets.

D. Constrained Portfolio Optimization (CPO)

To achieve a balanced portfolio that maximizes returns and minimizes risk, an investor with moderate risk aversion imposes specific preferences and conditions on the MVO model to construct the CPO model. For example, an investor wants 40% of the capital to be invested in high-volatility stocks and 60% in low-volatility stocks. Also, 1% of the capital must be invested in both classes, some assets may not be invested, and high-volatility stocks cannot exceed 10%. The weights add up to one, ensuring a full investment of available capital. The mathematical

formulation of the model is (see [15]):

$$\begin{aligned}
 \min \quad & - \sum_{i=1}^N W_i R_i + \sum_i \sum_j W_i W_j \sigma_{ij} \\
 \text{subject to} \quad & \\
 & \sum_{i=1}^N W_i = 1 \\
 & 0.01 \leq \sum_{i \in LowVolatility} W_i \leq 0.6 \\
 & 0.01 \leq \sum_{j \in HighVolatility} W_j \leq 0.4 \\
 & 0 \leq W_i \leq 1, \quad i \in LowVolatility \\
 & 0 \leq W_j \leq 0.1, \quad j \in HighVolatility.
 \end{aligned} \tag{3.3}$$

E. Sharpe Ratio Optimization (SRO)

The mathematical model for optimizing the Sharpe ratio based on the portfolio is (see [15]):

$$\begin{aligned}
 \max \quad & \left(\frac{\sum_{i=1}^N W_i R_i - R_f}{\sqrt{\sum_i \sum_j W_i W_j \sigma_{ij}}} \right) \\
 \text{subject to} \quad & \\
 & \sum_{i=1}^N W_i = 1, \quad 0 \leq W_i \leq 1
 \end{aligned} \tag{3.4}$$

In the objective function, the numerator represents the investment's excess returns compared to a risk-free asset, denoted as R_f , while the denominator represents the investment's risk. The goal is to maximize the Sharpe ratio. The basic constraints state that the investor wants to create a fully invested portfolio.

3.2.2 Data-Driven Portfolio Optimization

Traditional PO techniques typically overlook non-normality, skewness⁸, and kurtosis⁹ in the underlying dataset. In contrast, the data-driven portfolio construction method accounts for non-normality and skewness in financial data. Incorporating these statistical features enhances the robustness and practicality of the model.

Using sign and volatility correlations, two innovative volatility estimates were introduced in an earlier study by Thavaneswaran et al. [86]. These functions, characterized by smaller variances, are employed to formulate a novel risk measure for a portfolio that takes into account moments of higher order. The sign correlation ($\rho_{P,\text{sgn}}$) and volatility correlation ($\rho_{P,\text{vol}}$) are given as [86]:

$$\rho_{P,\text{sgn}} = \text{Corr}(\text{Sgn}(R_P - \mu_p), R_P - \mu_p)$$

and

$$\rho_{P,\text{vol}} = \text{Corr}(|R_P - \mu_p|, (R_P - \mu_p)^2).$$

Here, μ_p represents the expected return of the portfolio.

A. Risk Measures

Sign and volatility correlations are used by Bowala and Singh [85] to introduce four distinct risk measures. The portfolio's expected return is indicated by μ_p . R_P represents the portfolio return. The inverse of the cumulative distribution function (CDF)¹⁰ is denoted by $F(x)$ of portfolio return. $\tilde{R}_p$ and $\hat{\sigma}_P$ serve as estimates of the expected return and standard deviation (SD) of a financial portfolio, respectively, derived from the past l observations.

⁸Skewness measures the asymmetry of a data distribution. Positive skewness indicates a long right tail, while negative skewness indicates a long left tail.

⁹Kurtosis measures the “tailedness” of a data distribution. High kurtosis indicates heavy tails and more outliers, while low kurtosis indicates light tails and fewer outliers.

¹⁰The cumulative distribution function (CDF) of a random variable X is defined as the probability that X takes on a value less than or equal to a given point x . Formally, the CDF is given by $F(x) = P(X \leq x)$, where $F(x)$ represents the probability that X is less than or equal to x .

The mean absolute deviation (MAD) assessment for the portfolio is computed using [85]:

$$\text{MAD}_P = 2\hat{\rho}_{P,\text{sign}}\hat{\sigma}_P\sqrt{F(\bar{R}_P)(1 - F(\bar{R}_P))}.$$

The volatility estimate (VE) of the portfolio, employing volatility correlation (VEV), is quantified through [85]:

$$\text{VEV}_P = \sqrt{1 - \hat{\rho}_{P,\text{vol}}^2}\hat{\sigma}_P.$$

The VE of the portfolio utilizing sign correlation (VES) is given below [85]:

$$\text{VES}_P = \sqrt{1 - \hat{\rho}_{P,\text{sign}}^2}\hat{\sigma}_P.$$

For the volatility estimate of the portfolio considering both volatility and sign correlation reduction (VESV), the formula is as follows [85]:

$$\text{VESV}_P = \sqrt{(1 - \hat{\rho}_{P,\text{sign}}^2)(1 - \hat{\rho}_{P,\text{vol}}^2)}\hat{\sigma}_P.$$

The data-driven PO strategy considers these four distinct data-driven risk measures. Unlike other existing traditional PO techniques, the data-driven PO algorithm considers the underlying non-normality of the financial time series.

3.2.3 Diversification

We compute the diversification index by using the formula:

$$\frac{\bar{\sigma}\bar{W}'}{\sqrt{\bar{W}\Sigma\bar{W}'}} \quad (3.5)$$

where $\bar{\sigma}$ is the standard deviation vector, $\sqrt{\bar{W}}$ is the vector of weights, and Σ is the variance-covariance matrix of returns.

3.2.4 Clustering Approaches

This research employs four different clustering techniques to facilitate stock diversification within portfolios. Specifically, the less extensively explored methods of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and affinity propagation (AP) are studied for this thesis. The objective is to construct portfolios by selecting unique stocks from each cluster, thereby enhancing diversification strategies.

K-means clustering algorithm (see [87]): It is a prominent unsupervised ML approach utilized for partitioning a given dataset into K distinct clusters. This technique aims to minimize the collective sum of squared distances between individual data points and their corresponding cluster centroids. The Within Cluster Sum of Squares (WCSS) method is commonly employed to determine the suitable number of clusters.

Hierarchical Clustering Algorithm (see [87]) : It is an unsupervised ML algorithm that groups data points into hierarchical clusters, considering their similarity or dissimilarity. This method establishes a hierarchical arrangement of clusters, often represented as a dendrogram, by iteratively merging or splitting clusters. In our study, we leveraged the elbow plot technique, obtained by converting the dendrogram into a dissimilarity matrix. By utilizing the WCSS criterion, we ascertained the ideal cluster quantity essential for our clustering investigation.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (see [88]): It is an unsupervised ML algorithm that identifies clusters based on the concept of density. DBSCAN effectively employs two crucial parameters: epsilon (ϵ), which governs the maximum allowable distance between neighboring points, and minPts, which specifies the minimum number of points required to establish a dense region. The algorithm also adeptly detects noise or outlier points, which lie in regions of diminished density and fail to satisfy the density criteria. In our study, we employed the reachability distance metric to determine the optimal value of epsilon for the DBSCAN clustering algorithm.

Affinity Propagation (AP) (see [89]): It is an influential clustering algorithm that operates in an unsupervised manner that effectively identifies exemplars within a given dataset, allowing the data points to ‘vote’ on each other to determine their representative exemplar. The algorithm leverages the notion of message passing, wherein messages are exchanged between data points to ascertain the most suitable exemplar for each point. By iteratively updating these messages, AP converges to a set of exemplars that effectively represent the dataset. The robustness of the algorithm lies in its capacity to autonomously ascertain the cluster quantity

without necessitating prior knowledge. The versatility of this algorithm allows it to handle datasets with varying cluster sizes. This algorithm has been previously used in several applications [90] [91].

3.3 Data Science Concepts

Data science (DS) is a multidisciplinary field that integrates statistical methods, computational tools, and specialized knowledge to derive valuable insights and understanding from both structured and unstructured data. It includes the systematic application of algorithms, data processing methods, and machine learning to analyze large datasets, identify patterns, and support decision-making processes. DS encompasses a broad range of activities, including data cleaning, data integration, data visualization, and predictive modeling, aiming to transform raw data into actionable intelligence. In this thesis, for each study, a significant amount of work involves exploring the intricacies of the underlying complex financial data. Understanding the statistical features of the data helps in devising effective algorithmic strategies for price prediction and portfolio construction.

3.3.1 Data Pre-Processing

Data preprocessing for financial stock and cryptocurrency data is essential to ensure quality and reliability in analysis and modeling. The process begins with collecting historical and real-time data from reliable sources, including financial markets and exchanges. The next step, data cleaning, tackles problems such as missing values, outliers, and inconsistencies by applying statistical methods. Data transformation then reshapes raw data into an appropriate format, which includes scaling features and aligning time frames consistently. Feature engineering extracts meaningful features, enhancing model performance. Data integration combines multiple sources to form a comprehensive dataset, merging financial data. Finally, data reduction simplifies the dataset by reducing dimensionality through different statistical or ML approaches, retaining essential information while eliminating redundancy. This thorough preprocessing en-

hances the accuracy and efficiency of predictive models, facilitating better investment decisions through different PO techniques.

A. Data Inference

Data inference involves statistical methods and algorithms to extract meaningful insights and inform decision-making processes. In this research, financial data from stocks, stock indices, and cryptocurrencies are used, each having a different range of values. The initial step involves calculating returns from these raw values.

The formula for calculating stock returns from the closing price is:

$$\text{Return} = \frac{\text{Closing Price}_{\text{current}} - \text{Closing Price}_{\text{previous}}}{\text{Closing Price}_{\text{previous}}}. \quad (3.6)$$

Calculating stock returns from the closing price instead of using the prices directly provides a standardized measure of performance over time. Returns express the percentage change, accounting for the price scale and allowing meaningful comparisons across different periods and stocks, regardless of their absolute price levels. This method ensures a more accurate evaluation of investment performance and risk.

B. Standardization and Normalization

Standardization and normalization are crucial preprocessing techniques that enhance the efficacy of ML algorithms. Standardization modifies data to achieve a zero mean and a standard deviation of one, making features with different units or scales easily comparable.

The formula for standardization is as follows:

$$z = \frac{x - \mu}{\sigma}. \quad (3.7)$$

where:

- z is the standardized value,
- x is the original data point,
- μ is the mean of the data,

- σ is the standard deviation of the data.

Normalization adjusts data to fit within a specific range, usually between 0 and 1, aiding in managing data with different scales and enhancing the convergence of gradient-based algorithms. Both techniques are vital for ensuring that each feature has an equal impact on the analysis, preventing any single feature from overpowering the model due to its scale.

The formula for normalization is:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (3.8)$$

where:

- x' is the normalized value,
- x is the original data point,
- $\min(x)$ is the minimum value in the data set,
- $\max(x)$ is the maximum value in the data set.

Standardization and normalization are both techniques used to rescale data, but they serve different purposes and follow distinct methods. Standardization preserves the relative relationships between data points, making it suitable for algorithms that assume normally distributed data. Normalization adjusts data to fall within a defined range, typically from 0 to 1, which is beneficial for algorithms needing bounded input values or when dealing with features of varying scales. While standardization retains the data's distribution shape, normalization focuses on adjusting the data to a uniform scale.

C. Handling missing Data

Addressing missing values is an essential part of data preprocessing to maintain the quality and accuracy of the analysis. Common techniques include imputation, where missing values are substituted with estimates like the mean, median, or mode of the existing data.

The formula for mean imputation is:

$$\text{Imputed Value} = \frac{\sum_{i=1}^n x_i}{n}. \quad (3.9)$$

where:

- x_i are the observed values,
- n is the number of observed values.

The formula for median imputation is:

$$\text{Imputed Value} = \text{Median}(x_1, x_2, \dots, x_n). \quad (3.10)$$

where, $\{x_1, x_2, \dots, x_n\}$ are the observed values.

The formula for mode imputation is:

$$\text{Imputed Value} = \text{Mode}(\{x_1, x_2, \dots, x_n\}). \quad (3.11)$$

where, $\{x_1, x_2, \dots, x_n\}$ are the observed values.

Here,

- **Median:**

- If n is odd, the median is the $\left(\frac{n+1}{2}\right)^{\text{th}}$ data value.
- If n is even, the median is the average of the $\left(\frac{n}{2}\right)^{\text{th}}$ and $\left(\frac{n}{2} + 1\right)^{\text{th}}$ data values.

- **Mode:**

Mode = Most frequently occurring value in the data set

Alternatively, more advanced methods, such as using ML algorithms to predict missing values, can be utilized. In some instances, missing data points are completely removed if they are sparse and randomly distributed. Each technique has its benefits and is selected according to the characteristics of the data and the specific needs of the analysis. Effectively managing missing values helps avoid biases and enhances the reliability of the results.

3.3.2 Statistical Data Exploration

Statistical exploration involves analyzing and summarizing data to uncover patterns, trends, and relationships. This process generally involves computing descriptive statistics like the mean, median, and standard deviation, along with illustrating data using graphs and plots. By exploring data statistically, researchers can gain initial insights, identify anomalies, and formulate hypotheses. It also helps in understanding the underlying distribution and variability of the data, which is essential for selecting appropriate analytical methods and building reliable models. Statistical exploration is a foundational step in data analysis that sets the stage for more advanced techniques and decision-making. The studies in this thesis involve analyzing financial data from a statistical perspective to uncover significant features and underlying non-linear patterns.

A. Financial Data Distribution

Financial data distribution refers to how financial variables, such as asset prices, returns, and trading volumes, are spread across different values. Understanding the distribution is crucial for financial analysis, as it provides insights into the behavior and characteristics of the data. From the statistical exploration of the data distribution, this research identifies that financial data exhibit non-normal distributions, characterized by skewness, kurtosis, and heavy tails. Analyzing the distribution helps in identifying patterns, assessing risk, and making informed investment decisions. Techniques like histograms, probability density functions, and statistical tests are commonly used to study financial data distribution, revealing important aspects such as central tendency, variability, and the likelihood of extreme events.

3.3.3 Bollinger Bands

Bollinger Bands are a technical analysis tool designed to gauge market volatility and detect possible overbought or oversold conditions in financial markets. They include three lines: a central band, usually a 20-day simple moving average (SMA), and two outer bands that are standard deviations apart from the central band. The distance between the outer bands varies

based on market volatility, expanding during high-volatility periods and contracting during low-volatility periods. Bollinger Bands helps traders make decisions by providing visual indicators of price trends and potential reversal points. By observing how prices interact with these bands, traders can gain insights into whether an asset is overbought or oversold, potentially indicating a reversal or continuation of a trend.

Standard Bollinger Bands consist of the upper and lower bands, which are set two sample standard deviations above and below the middle band, respectively. Assuming a normal distribution, standard Bollinger Bands are defined as:

$$(Lower_{t,L}, Upper_{t,L}) = SMA_{t,L} \pm 2s_{t,L},$$

where,

$$s_{t,L} = \sqrt{\frac{\sum_{i=t-L+1}^t (d_i - SMA_{t,L})^2}{L}}$$

Here $s_{t,L}$ defines the standard deviation and d_i is the daily stock closing price. $Lower_{t,L}$ represents the lower band and $Upper_{t,L}$ represents the upper band.

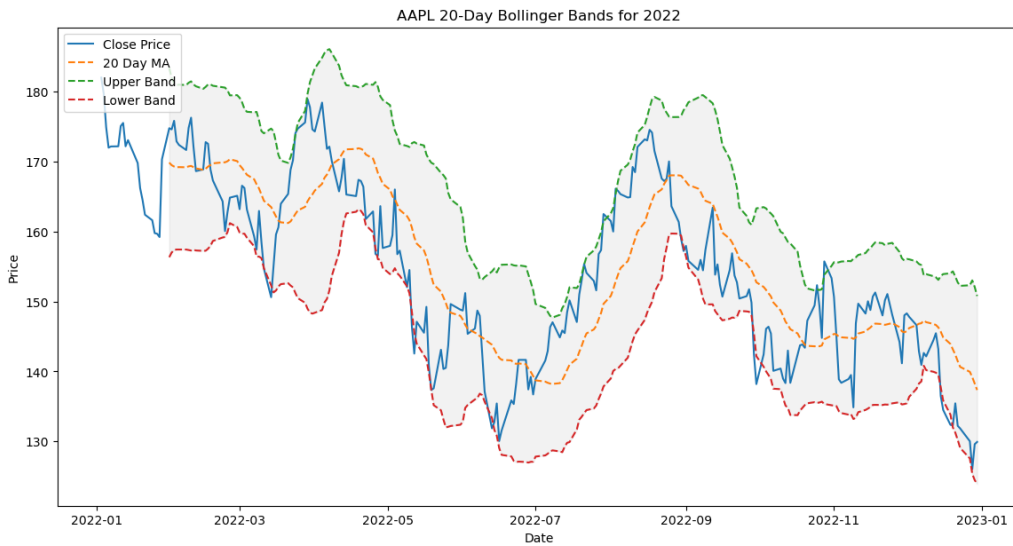


Figure 3.7: 20-day Bollinger Bands for 2022

Figure 3.7 shows a 20-day Bollinger band for 2022. The blue line indicates the closing price of Apple (AAPL) stock for 2022. The dashed orange central band represents the 20-day moving average (MA), calculated by taking the observations for 20 days and finding the mean of these

observations. The green dashed line is the upper band, calculated by finding the standard deviation of the past 20 days, multiplying it by 2, and adding it to the 20-day MA. Similarly, the red dashed line is the lower band, found by subtracting twice the standard deviation from the 20-day MA.

The width of the bands indicates the stock's volatility. When the bands are wide, it signifies high volatility, whereas narrow bands indicate low volatility. In the plot, there are periods where the bands widen significantly, such as around May and October, indicating higher volatility during these times.

3.3.4 Unsupervised Learning and Clustering Techniques

Unsupervised learning is a machine learning approach where the algorithm is trained on data that lacks explicit labels or predefined results. The primary goal is to uncover hidden patterns, structures, or relationships within the data. Clustering is a common unsupervised learning approach that groups similar data points into clusters based on their inherent characteristics. Popular clustering techniques include k-means clustering, which divides data into an exact number of clusters, and hierarchical clustering, which creates a tree-like structure of nested clusters. These methods are widely used in various fields, such as customer segmentation, image recognition, and anomaly detection, to discover meaningful patterns and insights from unlabelled data. Section 3.2.4 explains the clustering approaches used in this research.

Chapter 4

Experimentation

This chapter discusses the experimentation done with a variety of financial assets with proposed innovative models for asset price predictions and PO. The proposed solutions offer predictive accuracy and enhanced profitability under fluctuating market conditions and crisis scenarios. The experiments encompass creating efficient models for price prediction (see Sections 4.1,4.2), and enhancing profitability through introducing diversification for PO (see Sections 4.3, 4.4).

4.1 Encoder-Decoder Architecture for Financial Asset Price Prediction

This work addresses the intricate task of predicting the prices of diverse financial assets, including stocks, indices, and cryptocurrencies, each exhibiting distinct characteristics and behaviors under varied market conditions. To tackle the challenge effectively, novel encoder-decoder architectures, AE-LSTM and AE-GRU, integrating the encoder-decoder principle with LSTM and GRU, are designed. The experimentation involves multiple activation functions and hyperparameter tuning. With extensive experimentation and enhancements applied to AE-LSTM, the proposed AE-GRU architecture still demonstrates significant superiority in forecasting the annual prices of volatile financial assets from the multiple sectors mentioned above. Thus, the novel AE-GRU architecture emerges as a superior choice for price prediction

across diverse sectors and fluctuating volatile market scenarios by extracting important non-linear features of financial data and retaining the long-term context from past observations.

4.1.1 Dataset

The financial market encompasses a vast array of products derived from various sectors. Each financial product, whether it's a derivative of stocks, cryptocurrencies, *etc.*, reacts differently to distinct market events. Therefore, when developing a predictive model, it's crucial to consider testing under a variety of market conditions with these diverse financial derivatives. In this study, we have experimented with three primary financial products, from which multiple derivatives are formulated. These include volatile stocks from a range of sectors, the highly volatile cryptocurrency Bitcoin, and the moderately volatile yet stable stock index S&P. These are used for training and testing the proposed models.

To assess the effectiveness of the novel Autoencoder-GRU (AE-GRU) and compare its performance with Autoencoder-LSTM (AE-LSTM), this research has examined diverse stock datasets spanning various sectors. These include Apple (AAPL) (technology sector), Johnson and Johnson (JNJ) (healthcare sector), Chevron (CVX) (energy sector), JP Morgan Chase Bank (JPM) (banking sector), Bitcoin (BTC-USD) (cryptocurrency), and the S&P stock index (^GSPC). The models were trained on data from 1998 to 2022, and their predictive capabilities were evaluated using data from 2023. Additionally, for Bitcoin, the training dataset spans from 2014 to 2022, with the model tested on data from 2023. The dataset analyzed includes significant market occurrences, including the 2001 dot-com bubble burst, the 2008 global financial crisis, and the market crash induced by the COVID-19 pandemic in 2020.

The efficiency of the proposed architecture has been evaluated across multiple machines to analyze its performance under the influence of various machine configurations. These include (a) a Dell Latitude 5540 intel core i7-1355U, 1.70 GHz, with 10 Core(s) and 16GB RAM; (b) an ASUS VivoBook X515EA intel core i5-1135G7, 2.40 GHz, with 4 Core(s) and 16GB RAM; (c) DELL Inc. OptiPlex 7040, intel core i5-6600, 3.30 GHz, with 4 core(s) and 16 GB RAM.

All these machines run Windows 11 OS and Jupiter Notebook as the IDE.

4.1.2 Exploratory data analysis

Figure 4.1 depicts the Bollinger bands for the financial instruments analyzed in this study. Both Figure 4.1 and Table 4.1 collectively indicate that AAPL and JPM display elevated volatility, while JNJ demonstrates comparatively lower volatility, and CVX exhibits a moderate level of volatility. S&P registers notably low average volatility, whereas the cryptocurrency bitcoin experiences the highest volatility between 2022 and 2023 among all the assets considered.

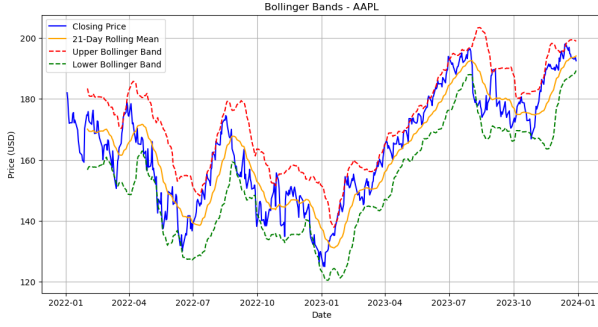
Table 4.1 illustrates the average volatility metrics derived from a 21-day rolling period for a range of assets, encompassing equities such as Apple, Johnson & Johnson, JP Morgan Chase, and Chevron, alongside the S&P index and the cryptocurrency Bitcoin. Each asset is associated with an average volatility value expressed in percentage terms, reflecting the average daily price oscillation observed over the specified 21-day timeframe.

For instance, Apple exhibits an average volatility of 35.96%, indicating that, on average, its stock price experiences a daily fluctuation of approximately 35.96% during the 21-day period. Similarly, the table furnishes comparable average volatility figures for other assets, facilitating assessments of their respective levels of price variability.

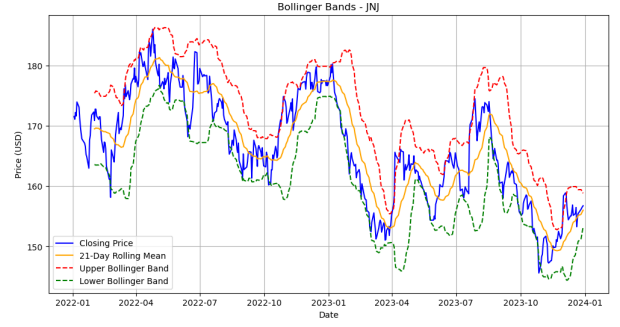
Table 4.1: Average Volatility based on 21 rolling days.

Assets	Average Volatility
Apple	35.96
Johnson & Johnson	17.60
JP Morgan Chase	31.28
Chevron	24.34
S&P	16.71
Bitcoin	52.48

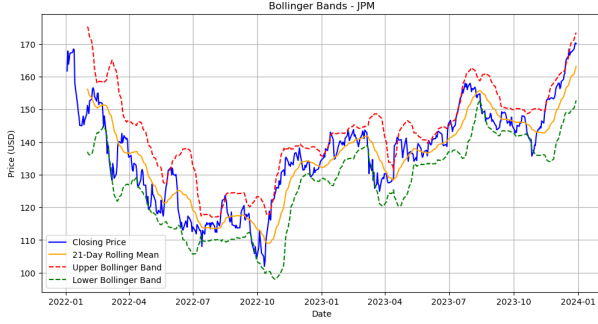
Figure 4.2 illustrates the change in the volatility trend in the considered timeframe. Notably, Apple stock exhibits a significant rise in volatility. Conversely, Bitcoin's volatility displays abrupt fluctuations. The S&P, reflecting a blend of diverse financial assets, exhibits a high magnitude of volatility but lower average volatility. On the other hand, JPM, JNJ, and CVX consistently experience growth in volatility throughout the years.



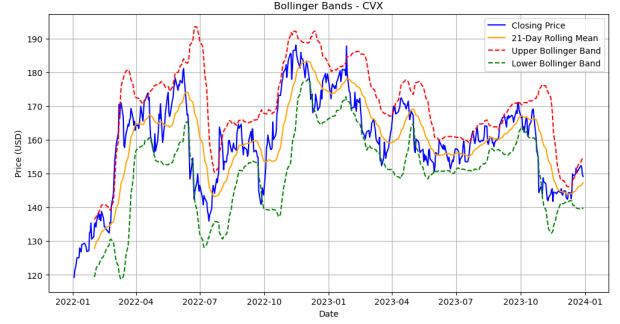
Bollinger plots for AAPL.



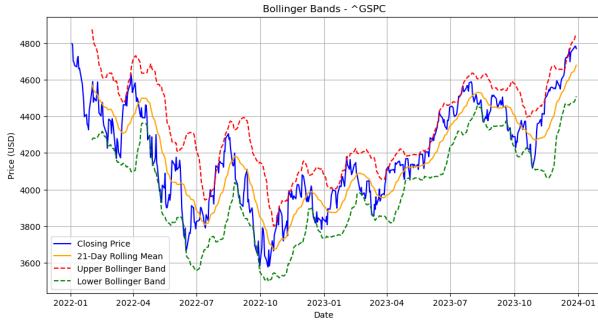
Bollinger plots for JNJ.



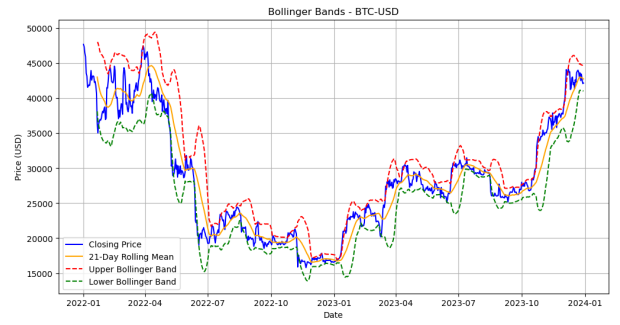
Bollinger plots for JPM.



Bollinger plots for CVX.



Bollinger plots for S&P.



Bollinger plots for BTC.

Figure 4.1: Bollinger plots for financial assets from 2022 to 2023.

4.1.3 Implementation

A. Stacked AE-LSTM and AE-GRU architecture

As discussed in the chapter 3.1.4, an autoencoder is a neural network architecture designed for unsupervised learning, consisting of an encoder and decoder. It aims to learn a compact representation of input data by reducing the reconstruction error between the original and reformed data. The proposed models in this research utilized the features of an stacked autoencoder.

In the realm of financial time series forecasting, the process of extracting non-linear features

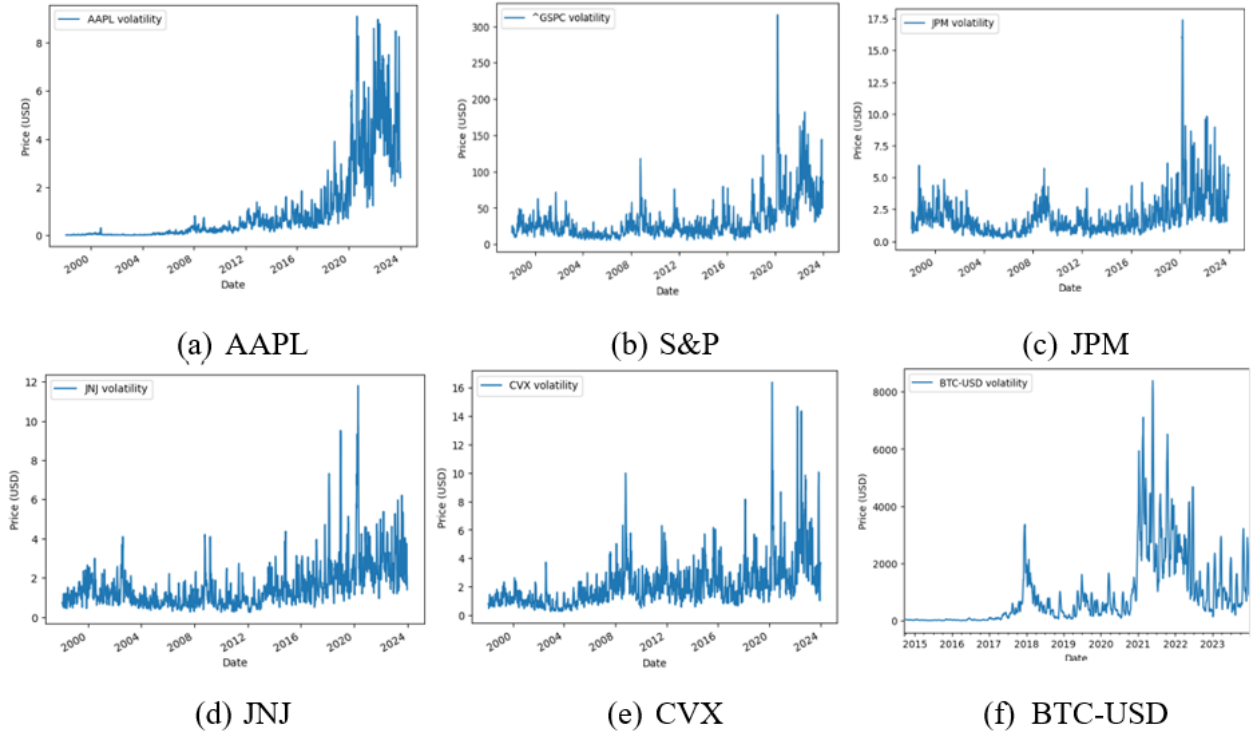


Figure 4.2: Change in volatility.

from a dataset is of paramount importance. Financial time series data is unique in that it often exhibits non-linear characteristics. This means that the relationships between variables in the data are not simply proportional, but can change in complex ways. These non-linear relationships can be influenced by a multitude of factors, making them challenging to model accurately.

Algorithms designed for prediction tasks, such as those used in machine learning, are capable of learning from these inherent patterns within the data. They do this via a process called training, which adjusts model to minimize the difference between their predictions and the actual data. Over time, this training process allows the algorithm to improve its predictive accuracy.

LSTM and GRU are commonly employed in the prediction of stocks and cryptocurrencies. However, these basic LSTM and GRU structures do not take into account the significance of non-linear features. In NLP, both LSTM and GRU encounter difficulties in predicting words when the sentence length increases significantly. Consequently, these algorithms struggle to

remember the context over a very long term. To address this issue, the transformer network (TN) was introduced, which uses an attention mechanism in an encoder-decoder architecture to remember the long-term context [84]. However, the encoder-decoder architecture of TN doesn't compress the input vector for learning intricate features from the latent representation. The encoder component of the Transformer handles the processing of the input sequence, while the output sequence is produced by the decoder. However, the implementation of such a complex model is computationally demanding. In the prediction of financial time series, it is crucial to remember the long-term context of past significant market events. This aids in identifying similar trends and predicting future values based on these trends. Therefore, the basic LSTM and GRU architecture is not the optimal solution for this task. Typically, for a one-day-ahead forecast for stocks and cryptocurrencies, the training dataset is usually very large. Consequently, running such a large dataset with a TN architecture becomes computationally prohibitive. Encoder-decoder-based RNN architectures demonstrated effective performance when an Autoencoder-based LSTM was proposed by Rubell et al. [38]. To a certain degree, the encoded latent representation of the input data preserved the long-term contextual information about past observations, which helped to enhance the predictive accuracy of the basic LSTM architecture. In this study, we found that significant improvements in predictive accuracy could be achieved through extensive experimentation with the model's hyperparameters. However, in some studies, the GRU outperformed the LSTM in terms of achieving higher predictive accuracy [92], [93]. Therefore, this research also examines the accuracy of such an encoder-decoder architecture with GRU and proposes a new model with improvised performance.

This research introduces innovative frameworks comprising stacked Autoencoder-based LSTM (AE-LSTM) and GRU (AE-GRU). The foundational structure of the proposed algorithm is elucidated in Figure 4.3. While both AE-LSTM and AE-GRU share a common architecture, distinct hyperparameters and activation functions tailored to each architecture were employed for optimal performance after strenuous investigation.

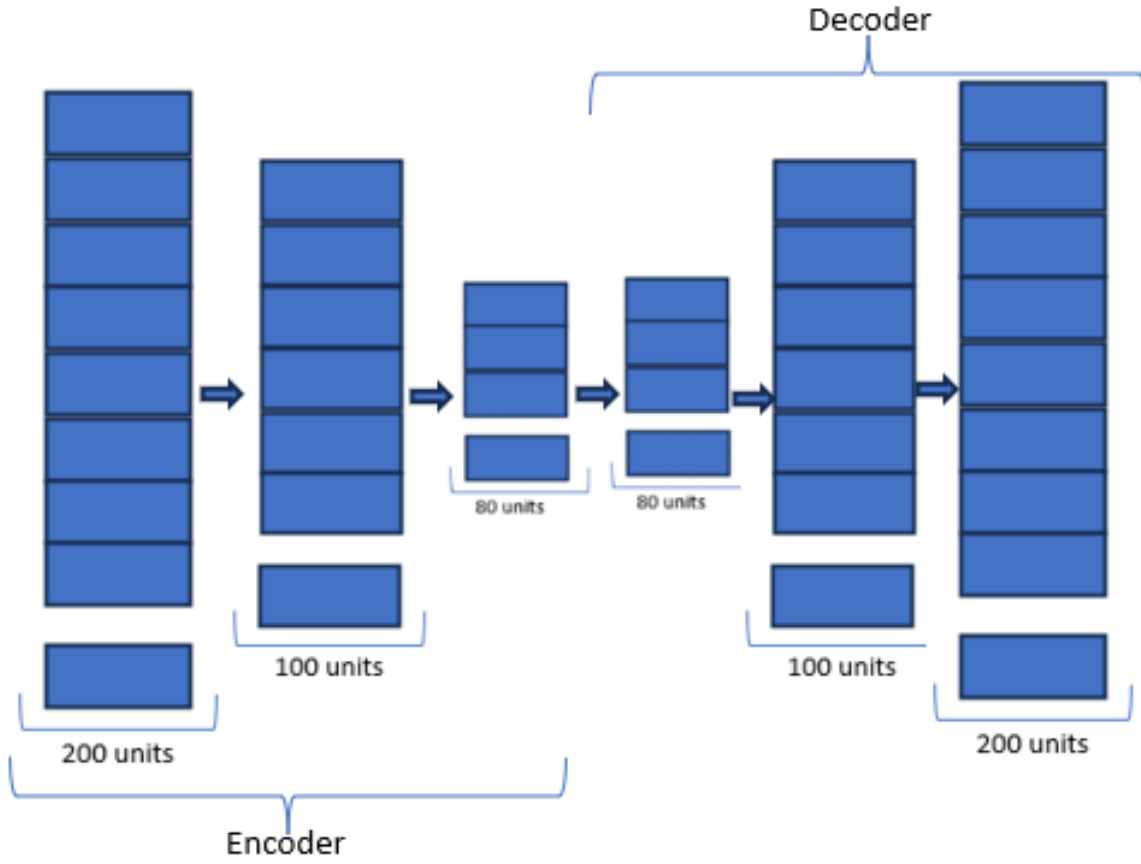


Figure 4.3: Encoder-decoder-based novel LSTM and GRU architecture.

AE-LSTM: AE-LSTM, or Autoencoder-based LSTM, is a hybrid neural architecture that combines the capabilities of an autoencoder with LSTM units. Autoencoders are employed to learn a compressed representation of input data, and this encoded information is then fed into LSTM layers, enabling the model to capture and leverage long-term dependencies in sequential data. AE-LSTM is particularly effective in tasks involving time-series data, such as stock price prediction, where both feature learning and sequential context preservation are crucial for accurate forecasting. Figure 4.3 represents the model structure of the proposed improvised AE-LSTM architecture. The encoder-decoder architecture of AE-LSTM can be expressed as the following data-flows:

$$\text{Encoder: } x \xrightarrow{\text{LSTM } 200} h_1 \xrightarrow{\text{LSTM } 100} h_2 \xrightarrow{\text{LSTM } 80} z,$$

$$\text{Decoder: } z \xrightarrow{\text{LSTM } 80} h_3 \xrightarrow{\text{LSTM } 100} h_4 \xrightarrow{\text{LSTM } 200} \hat{x}.$$

Given the data-flows above, it's evident that the encoder structure employs several stacked LSTM layers to identify latent features at various compression levels. With each level of compression, the architecture learns more prominent features. However, additional compression leads to information loss. Such a deep architecture results in a higher reconstruction loss and is susceptible to the vanishing gradient problem. The initial compression occurs when the number of units in the second LSTM layer is reduced from 200 to 100. Over time, each layer experiences a reduction in units until the latent representation z is finally extracted from the LSTM layer with 80 units.

Similarly, the decoder layer incrementally increases the units in the subsequent LSTM layers. The entire structure is trained by minimizing the reconstruction loss. The output of the decoder is the reconstructed data that encapsulates the internal non-linear complexities of the financial time series. Ultimately, a dense layer is employed to obtain the predicted output.

Algorithm 3: AE-LSTM

```

Data:  $data \leftarrow parse("stock\_name")$ 

 $normalized\_data \leftarrow normalize(data);$ 

 $train\_data \leftarrow normalized\_data[start : end];$ 

 $test\_data \leftarrow normalized\_data[start : end];$ 

 $preprocessed\_train \leftarrow preprocess(train);$ 

 $encoder \leftarrow LSTM(units = 200, activation = 'activation')(preprocessed\_train);$ 

 $encoder \leftarrow LSTM(units = 100, activation = 'activation')(encoder);$ 

 $encoder \leftarrow LSTM(units = 80, activation = 'activation')(encoder);$ 

 $decoder \leftarrow LSTM(units = 80, activation = 'activation')(encoder);$ 

 $decoder \leftarrow LSTM(units = 100, activation = 'activation')(decoder);$ 

 $decoder \leftarrow LSTM(units = 200, activation = 'activation')(decoder);$ 

 $model \leftarrow Model(encoder, decoder);$ 

 $model.compile();$ 

 $predict \leftarrow model.predict(test\_data);$ 

```

Algorithm 3 shows an algorithmic view of the proposed novel improvised encoder-decoder architecture of LSTM which outperforms the existing models.

AE-GRU: AE-GRU is a hybrid neural network architecture that combines the capabilities of an autoencoder with a GRU. This innovative model integrates the feature learning abilities of autoencoders with the sequential modeling strengths of GRU, aiming to enhance the representation and prediction capabilities in various applications such as time series analysis and sequential data processing. The AE-GRU architecture involves the use of a stacked autoencoder for feature extraction, feeding the encoded features into GRU cells for capturing temporal dependencies and patterns. This combination leverages both unsupervised learning and RNN architectures to achieve improved performance in tasks requiring sequential data understanding and prediction. To our knowledge, this hybridization is a novel approach to financial asset price prediction. Figure 4.3 represents the model structure of the proposed novel AE-GRU algorithm. AE-GRU architecture can be expressed as the following expressions:

$$\text{Encoder: } x \xrightarrow{\text{GRU } 200} h_1 \xrightarrow{\text{GRU } 100} h_2 \xrightarrow{\text{GRU } 80} z,$$

$$\text{Decoder: } z \xrightarrow{\text{GRU } 80} h_3 \xrightarrow{\text{GRU } 100} h_4 \xrightarrow{\text{GRU } 200} \hat{x}.$$

In the above expressions, we can observe that AE-GRU follows a similar architecture as AE-LSTM. Multiple encoder layers are layered to condense the feature vector into a representation in the latent space. After compressing to a feature vector, z , with an GRU layer consisting of 80 units, it is observed that further compression leads to a loss of information.

Several GRU layers, each with varying units, are stacked in a descending order to construct the encoder component of the AE-GRU architecture. This compression into the latent space z preserves valuable non-linear features of the financial time series. This latent feature vector encapsulates the long-term context derived from past observations.

Similarly, the decoder component incrementally increases the units in the GRU layers, resulting in an accurate reconstruction of the input data. This process eliminates outliers and extracts the non-linear contextual features from the financial data. Subsequently, a dense layer comprising a single unit is employed to predict future values.

Algorithm 4: AE-GRU

```

Data:  $data \leftarrow \text{parse}(\text{"stock\_name"})$ 

 $normalized\_data \leftarrow \text{normalize}(data);$ 

 $train\_data \leftarrow normalized\_data[start : end];$ 

 $test\_data \leftarrow normalized\_data[start : end];$ 

 $preprocessed\_train \leftarrow \text{preprocess}(train);$ 

 $encoder \leftarrow GRU(units = 200, activation = 'activation')(preprocessed\_train);$ 

 $encoder \leftarrow GRU(units = 100, activation = 'activation')(encoder);$ 

 $encoder \leftarrow GRU(units = 80, activation = 'activation')(encoder);$ 

 $decoder \leftarrow GRU(units = 80, activation = 'activation')(encoder);$ 

 $decoder \leftarrow GRU(units = 100, activation = 'activation')(decoder);$ 

 $decoder \leftarrow GRU(units = 200, activation = 'activation')(decoder);$ 

 $model \leftarrow \text{Model}(encoder, decoder);$ 

 $model.compile();$ 

 $predict \leftarrow model.predict(test\_data);$ 

```

Algorithm 4 shows an algorithmic view of the proposed novel encoder-decoder architecture of GRU, which outperforms the existing models and even the improvised AE-LSTM designed in this study. This encoder-decoder-based GRU architecture is novel and tested rigorously under varying market conditions. This proposed novel model overperforms existing architectures by retaining long-term non-linear contextual features.

B. Hyperparameter Tuning

The process of tuning hyperparameters is a critical step in the development of ML or DL algorithms. Hyperparameters are distinct from other parameters in that their values are determined before the learning process begins. These are not derived from the data during training but are manually set to guide the learning process. The selection of these hyperparameters significantly influences the functioning of the algorithm. Incorrect choices could lead to sub-optimal performance, even for a model with a high potential for asset price prediction.

Number of Layers: This study is done by experimenting with various configurations of encoder and decoder architectures and revealed that while multiple layer configurations were explored, exceeding a depth of three layers led to a notable deterioration in model performance. In such instances, the efficacy of the model was significantly compromised. Thus, after thorough experimentation, we have kept three layers in both the encoder and decoder. Increasing the number of layers increases the complexity of optimization.

Number of Units: Different numbers of units are also used for the encoder and decoder architecture. But the combination of units in the order illustrated in Figure 4.3 yields in the best accuracy.

Activation Functions: In major financial research, three activation functions are most used in training the DL models. These are: hyperbolic tanh, rectified linear unit (ReLU), and exponential linear unit (ELU). Combinations of different activation functions have been done for each category of financial assets in this study. Details about the activation functions can be obtained from chapter 3.1.3.

Table 4.2: Activation functions for different financial assets in AE-LSTM and AE-GRU models.

Financial Asset	AE-LSTM Activation Function	AE-GRU Activation Function
General Stocks	ELU	ReLU
Stock index	ReLU	tanh
Cryptocurrency	ELU	ReLU

After hyperparameter tuning, from Table 4.2, it can be observed that different activation functions perform best for different financial assets. For general stocks across various categories, the proposed AE-LSTM model demonstrates its maximum efficacy when employing the ELU as the activation function. Conversely, the ReLU exhibits superior performance when integrated into the AE-GRU architecture. When focusing on relatively stable stock indices, such as the S&P, AE-LSTM exhibits improved performance with ReLU activation, while AE-GRU demonstrates higher accuracy levels with the tanh activation function. In the case of highly volatile cryptocurrencies, like Bitcoin, Table 4.2 illustrates that AE-LSTM, utilizing ELU activation, achieves maximum efficiency, whereas AE-GRU, with ReLU activation, displays effective predictive accuracy. Thus, it becomes evident that the activation functions selected for the proposed

architectures exhibit similarities for moderately volatile stocks and highly volatile cryptocurrencies. This observation underscores the importance of hyperparameter tuning with respect to activation functions, revealing that the choice of activation function is contingent upon the inherent volatility of the financial assets under consideration.

Optimizers: Optimizer is a crucial algorithmic step to optimize the ML and DL algorithms so that the cost function is reduced, and proper weights are selected through backpropagation. In this study, we have experimented with 4 major optimizers.

These are:

1. Stochastic Gradient Descent (SGD): For this optimizer, the hyperparameters considered for optimization are learning rate and momentum.
2. Adagrad: For this optimizer, only the learning rate was considered for hyperparameter tuning.
3. RMSprop: For this optimizer, the learning rate and the decay rate were considered for hyperparameter tuning.
4. Adam: For this optimizer, learning rate, first and second-moment exponential decay rates are considered for hyperparameter tuning.

After extensive hyperparameter tuning conducted via grid search, the Adam optimizer showcased the highest predictive efficacy. This was achieved when configured with a learning rate, denoted by the symbol α , set to 0.001, a first-moment exponential decay rate represented by β_1 at 0.9, and a second-moment exponential decay rate, denoted by β_2 , set to 0.999. These configurations were found to be optimal across various categories of financial assets.

Epochs and Batch size: This study has selected various values of epochs for experimentation, including 60, 70, 100, 200, and 300 iterations. Following hyperparameter tuning through grid search, the results indicated that selecting epochs as 100 yielded the highest predictive accuracy. When the epochs are selected as 200, the predictive accuracy slightly differed in one machine. Similarly, among multiple values considered for batch size, hyperparameter tuning suggested using 32 and 64 as optimal batch sizes.

Time window: In pursuit of fortifying the model’s resilience, this study has undertaken an examination of three discrete time windows. These windows serve as the basis for training the model to forecast values one day ahead. Specifically, experimentation encompassed time windows of 60, 80, and 120 days, respectively. Notably, the empirical findings underscore the superior performance achieved within the 120-day time window.

Consider a scenario where each general stock undergoes trading for 252 days within a calendar year. For this illustration, let us designate the time window as comprising 120 days. In this context, if we utilize data spanning from the 1st day to the 120th day as input, the model will generate a prediction for the closing price on the 121st day. Subsequently, in the next iteration, with data spanning from the 2nd day to the 121st day, the model will furnish a prediction for the closing price on the 122nd day.

Grid search serves as a method for fine-tuning crucial parameters, particularly in the context of hyperparameter optimization. It operates by systematically traversing a predefined grid of hyperparameter combinations in pursuit of the most advantageous set. This method entails exhaustively assessing each combination, rendering it computationally demanding, particularly when dealing with expansive hyperparameter spaces. To ensure thorough evaluation, grid search commonly integrates k-fold cross-validation, dividing the dataset into subsets and scrutinizing each combination across multiple folds. The hyperparameter configuration yielding the highest cross-validation score is deemed the optimal set for the model, furnishing a robust basis for model performance assessment. However, a notable limitation of the grid search technique is its protracted execution duration.

In this context, grid search is specifically employed to identify optimal activation functions and hyperparameters on various optimizers, batch size, epochs, and number of units. Hyperparameter tuning through grid search is highly resource-intensive and execution takes a huge amount of time. During the execution of grid search without GPU-based resources, it took more than 24 hours to train a model for one single financial asset.

C. Evaluation Metrics

This research has tested the effectiveness of the models from five distinct viewpoints using five different evaluation metrics. Each of these metrics provides a unique way to test the models. In addition to the four evaluation metrics detailed in chapter 4.2.2, this study also incorporates the Root Mean Squared Error (RMSE).

Root Mean Squared Error (RMSE): It serves as a performance metric in regression analysis, measuring the mean measure of discrepancies between forecasted and true values. More precisely, it is derived from computing the square root of the MSE. RMSE is given by

$$\text{RMSE} = \sqrt{\text{MSE}}.$$

To summarize, this research has considered three major activation functions—ReLU, tanh, and ELU. Other hyperparameters include 1st-moment exponential decay rate β_1 , 2nd-moment exponential decay rate β_2 , learning rate, and numerical stability. The models take 120 previous time-step data to predict the next step. The predictive accuracy of the models developed in this study exceeds that of existing models for financial price predictions. The models proposed in this research demonstrate superior performance across various types of financial assets, spanning diverse market conditions. The outcomes of these model experiments are elaborated upon in Chapter 5.1.

4.2 Hybrid Data-Driven DL-based PO

This study introduces a novel hybrid architecture that combines deep learning, data-driven algorithms, and an AP-based approach to build robust investment portfolios. It evaluates the efficacy of BiLSTM and BiGRU in constructing resilient portfolios of stocks from diverse sectors under varying market conditions. The results highlight the superior performance of BiGRU, particularly in dynamic and volatile market scenarios. The research emphasizes the importance of precise stock prediction and effective diversification for building resilient portfolios, leveraging advanced techniques from deep learning and data-driven optimization. Comparative analyses

indicate similar performance between portfolios constructed with actual and predicted data using data-driven optimization. The findings offer valuable insights into constructing robust portfolios by employing advanced techniques, thereby enhancing decision-making in financial markets.

4.2.1 Dataset

In this study, a diversified portfolio comprising stocks from five distinct sectors—technology, healthcare, energy, banking, and consumer—along with two notable cryptocurrencies, Bitcoin (BTC-USD) and Ethereum (ETH-USD), has been meticulously selected. Each sector is represented by three carefully chosen stocks. The selected stocks include prominent entities such as Apple (AAPL), Microsoft (MSFT), Tesla (TSLA), Johnson & Johnson (JNJ), Pfizer Inc. (PFE), Merck & Co., Inc. (MRK), Exxon Mobil Corporation (XOM), Chevron Corporation (CVX), Enbridge Inc. (ENB), JPMorgan Chase & Co. (JPM), Bank of America Corporation (BAC), HSBC Holdings plc (HSBC), The Procter & Gamble Company (PG), The Coca-Cola Company (KO), and Unilever PLC (UL).

The dataset for stock analysis encompasses information spanning the years 1998 to 2022, serving as the training data. This temporal range encapsulates pivotal financial events, including the www crash of 2000, the global financial crisis of 2008, and the 2020 COVID-19-induced financial crisis. Additionally, minor yet impactful financial events within this timeframe are also considered. Leveraging this extensive training data, the study formulates predictions for the year 2023, subsequently utilizing the predicted data for portfolio optimization.

For Bitcoin, the training data spans from 2014 to 2022, while Ethereum’s training dataset covers the period from 2017 to 2022. This comprehensive approach enables the study to incorporate critical financial events, fostering a robust analysis for predictive modeling and portfolio optimization in the cryptocurrency domain.

4.2.2 Implementation

A. Exploratory Analysis

The dataset considered for this research is from 1998 to 2023. This dataset captures all major and minor events in this timeframe. Thus the BiLSTM and BiGRU models learn from the patterns of the drastic market events. This section explores interesting market intricacies especially from 2019 to 2023.

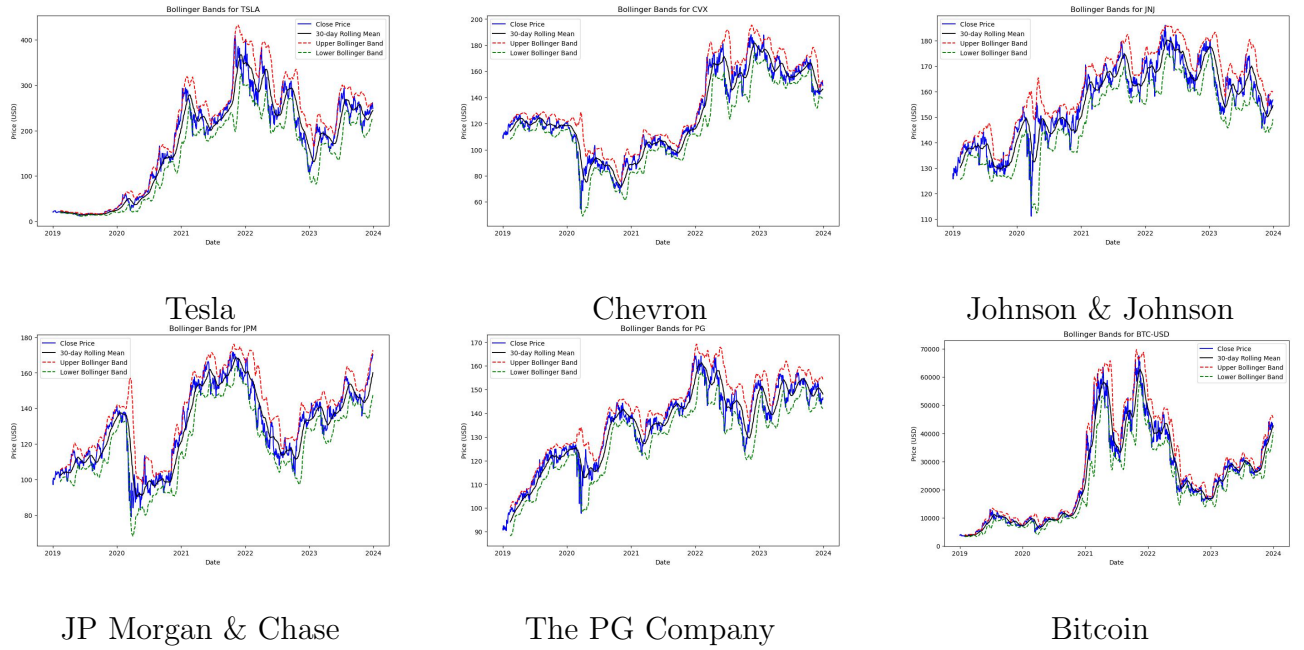


Figure 4.4: Bollinger bands for stocks and cryptos from 2019 to 2023

Examining Figure 4.4 and Table 4.3, it becomes evident that Tesla (TSLA) exhibits the highest level of volatility. In addition to Tesla (TSLA), Chevron (CVX) demonstrates notable volatility, albeit not to the extent observed in Tesla. Beyond traditional stocks, Bitcoin (BTC) manifests the utmost volatility, attributed to the inherently unpredictable nature of cryptocurrencies. Forecasting the performance of fluctuating assets proves challenging, primarily due to their pronounced volatility. Consequently, the formulation of portfolios incorporating highly unpredictable cryptocurrencies becomes a formidable task.

The visual representation in Figure 4.4 indicates that throughout the COVID-19 pandemic in 2020, the general market performance experienced a downturn, excluding certain technology stocks such as TESLA. Both Tesla and Bitcoin also displayed a gradual increase in their prices

Table 4.3: Volatility with 20 days rolling window

Components	Volatility
TSLA	108.30
JNJ	14.89
JPM	20.01
CVX	30.96
PG	16.65
BTC	16064.68

during this period. After the conclusion of the COVID-19 era, the lingering impact of the financial crisis persisted, leading to noticeable price fluctuations in 2022.

B. Model Design

The proposed novel PO framework depicted in Figure 4.5 comprises three main components. First, it includes a DL component utilizing pre-trained BiLSTM and BiGRU models for stock and cryptocurrency prediction. Second, it incorporates the AP algorithm for selecting stocks for portfolio construction. Third, it integrates a data-driven algorithm for optimizing portfolio weights. As indicated by the results from the experiments in Section 5.4, the AP algorithm combined with the data-driven portfolio optimization approach demonstrates maximum profitability through enhanced diversification. Therefore, the proposed model incorporates AP to leverage diversification benefits.

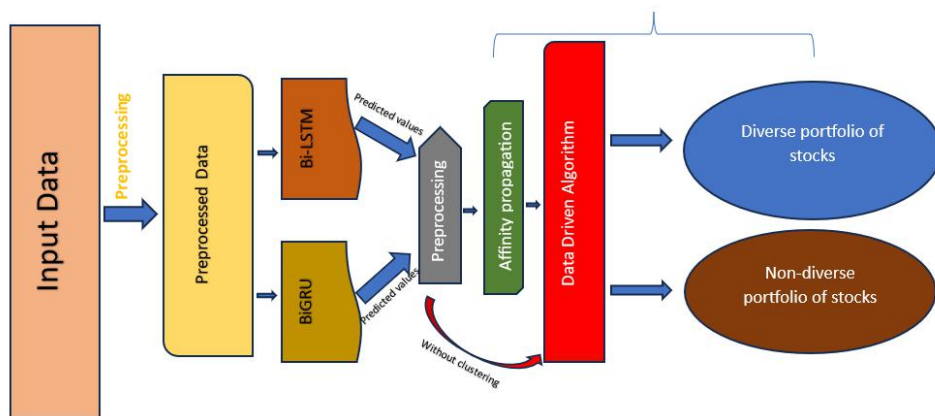


Figure 4.5: Proposed Algorithmic Architecture for Portfolio Optimization

Figure 4.5 illustrates the proposition of a novel integration involving the architecture of BiLSTM (see Section 3.1.1) and BiGRU (see Section 3.1.2). This integration is employed for

predicting future financial asset values, followed by diversification in stock selection using AP-based clustering. Subsequently, a data-driven PO algorithm is utilized for the construction of robust portfolios. In other words, this research introduces a novel framework that combines DL and a data-driven PO algorithm. This innovative approach incorporates the infrequently utilized AP-based clustering for diversification, aiming to create a resilient PO technique with a focus on risk aversion.

The input data consists of raw stock data, which undergoes preprocessing to handle missing/null values. Subsequently, returns are calculated and the data is normalized before being fed into the BiLSTM and BiGRU frameworks for prediction. The predicted values undergo further preprocessing, and a covariance matrix is computed. This matrix is then utilized in AP to identify clusters. Assets with the highest mean returns from each cluster are selected for inclusion in the portfolio. These selected stocks are then subjected to weight optimization using a data-driven algorithm.

Additionally, all stocks are included in a portfolio where weight optimization is conducted without incorporating diversity through AP. This comparison aims to evaluate how a diversified portfolio identified through cluster analysis outperforms a non-diversified portfolio in terms of profitability.

C. Evaluation Metrics

For quantifying model performance, this study has explored four popular evaluation metrics. These metrics are used to quantify how well the deep learning models are in forecasting future values.

Mean Squared Error (MSE): It assesses the average squared discrepancy between the observed values and the true values in a dataset. It is given by

$$\text{MSE} = \frac{1}{m} \sum_{j=1}^m (b_j - \hat{b}_j)^2;$$

here b_j is the original price and the $\hat{b}_j$ is the forecasted value.

Mean Absolute Error (MAE): It quantifies the mean extent of errors between forecasted

and genuine values in a dataset. Mathematically,

$$\text{MAE} = \frac{1}{m} \sum_{j=1}^m |b_j - \hat{b}_j|$$

where the number of observations in the dataset is represented by m . The true value for observation j is represented by b_j . The predicted value for observation j is represented by $\hat{b}_j$.

R-Square: In a regression model, the R-squared metric quantifies the ratio of the variance in the dependent variable that the independent variables account for. Mathematically,

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}}$$

where SSR is the sum of squared discrepancies between the observed values and the predicted values and SST is the total sum of squared discrepancies between the observed values and the average of the observed values.

Mean Absolute Percentage Error (MAPE): It serves as a measure to evaluate the precision of a forecasting model by determining the mean percentage disparity between anticipated and real values. Its significance lies in appraising model performance across varied datasets, furnishing a standardized gauge of predictive accuracy. Mathematically,

$$\text{MAPE} = \frac{1}{m} \sum_{j=1}^m \left| \frac{A_j - P_j}{A_j} \right| \times 100.$$

The number of observations in the dataset is represented by m . A_j represents the actual value for observation j . P_j represents the predicted value for observation j .

4.2.3 Portfolio Construction and Optimization

In this experiment, the proposed model explored both BiLSTM and BiGRU for financial prediction. Subsequently, weight optimization was conducted with and without introducing diversification. The use of diversification through the AP clustering algorithm resulted in a profitable portfolio comprising diverse financial assets. Previous experiments demonstrate that diversification enhances resilience and maximizes profitability in portfolio optimization techniques.

The data-driven PO approach was chosen for this proposed architecture because it incorporates crucial statistical features of the underlying financial dataset, which traditional approaches

often overlook. This enhances the practicality of the model presented in this research. Detailed results of these experiments are discussed in Chapter 5.2.

4.3 Resiliency in PO with Clustering technique

Constructing resilient portfolios is of crucial and utmost importance to investment management. This study compares traditional and data-driven models for building resilient portfolios and analyzes their performance for stocks (S&P 500) and highly volatile cryptocurrency markets. The study investigates the performance of traditional models, such as mean-variance and constrained optimization, and a recently proposed data-driven resilient portfolio optimization model for stocks. Moreover, the study analyzes these methods with evolving S&P CME bitcoin futures index and the Crypto20 index. These analyses highlight the need for further investigation into traditional and data-driven approaches for resilient portfolio optimization, including higher-order moments, particularly under varying market conditions. This study provides valuable insights for investors and portfolio managers aiming to build resilient portfolios that could be used in different market environments.

4.3.1 Dataset

This study considered twenty-five popular technological stocks from the S&P 500 index and twenty-five cryptocurrencies, and the selection of cryptocurrencies is based on their market capital (<https://coinmarketcap.com/>) (Table 4.4). Four years of data (2019 to 2022) for technological stocks have been collected for analysis. Some of the selected cryptocurrencies (e.g., Serum) did not start trading until 2022. Thus, this study considered cryptocurrency data for ten months of 2022 (January to October). This also allowed investigation of the performance of the traditional and data-driven approaches with the observation of limited highly volatile crypto data. All the adjusted closing prices for stocks and cryptocurrencies are downloaded from Yahoo!Finance¹. In the case of CPO, beta measurement is calculated with the corre-

¹<https://ca.finance.yahoo.com/>

sponding index. S&P 500 is a popular and well-established index for stocks. However, such an index is not yet available for crypto, and thus, this study considered two still evolving crypto-indexes, S&P CME Bitcoin Futures Index (Bitcoin and Ethereum-based index)² and Crypto20³ (considers top 20 cryptocurrencies), and their performances have been studied in portfolio optimization.

Table 4.4: Stocks and Cryptocurrencies

Regular Stocks	Cryptocurrencies
Apple Inc., Accenture plc, Adobe Inc., Automatic Data Processing Inc., Applied Materials Inc., Broadcom Inc., salesforce.com Inc., Cisco Systems Inc., Fidelity National Information Services Inc., Alphabet Inc. (Class C), Alphabet Inc. (Class A), International Business Machines Corporation, Intel Corporation, Intuit Inc., Facebook, Inc., Microsoft Corporation, Micron Technology Inc., Netflix, Inc., NVIDIA Corporation, Oracle Corporation, PayPal Holdings Inc., QUALCOMM Incorporated, Tesla, Inc., Texas Instruments Incorporated, Verizon Communications Inc.	Bitcoin, Cardano, Binance USD, Tether, Solana, Algorand, Terra Luna, Shiba Inu, Binance Coin, Avalanche, Chain Link, Serum, Bitcoin Cash, File Coin, Ethereum Classics, Polygon, Aave, Tron, Stellar, Vchain, Elrond, Litecoin, Ethereum, Ripple, Dogecoin.

It can be observed from Table 4.5 of selected stocks and cryptocurrencies that they have non-normal distributions with heavy-tailed skewness and high kurtosis, necessitating the integration of higher-order moments in risk measures.

Table 4.5: Summary Statistics - Stocks and Cryptocurrencies

Asset	Mean	SD	ρ_{vol}	ρ_{sign}	Skewness	Kurtosis
Apple	0.0023	0.0215	0.8877	0.6881	-0.0280	6.0019
Adobe	0.0018	0.0218	0.8395	0.6687	0.2317	10.7075
Automatic	0.0009	0.0193	0.8761	0.6250	-0.5128	13.6509
Bitcoin	-0.0026	0.0378	0.9150	0.7150	-0.1431	2.8213
Tether	-0.0000	0.0004	0.9361	0.4065	-6.8235	84.3960
Solana	-0.0085	0.0488	0.9208	0.5422	-3.3065	20.5880

²<https://www.spglobal.com/spdji/en/indices/digital-assets/sp-cme-bitcoin-futures-index/#overview>

³<https://Crypto20.com/en>

4.3.2 Stock and Cryptocurrency Selection through Clustering

A. Clustering for stock selection

This study uses clustering for stock selection to diversify investments. Technology stocks generally have high mean returns, except in 2022 where negative returns are observed. A feature matrix was used for clustering, with 12 clusters chosen for each year based on WCSS plots. K-means clustering was used to select assets for two portfolios, K-01 and K-02, based on the highest and second-highest mean returns from each cluster. Clustering varies yearly due to different mean returns and covariance matrices for the feature matrix. Table 4.6 captures a partial list of stock assets in each portfolio in each year.

Table 4.6: Clusters with mean returns

Year	K-01	K-02
2019-2020	'Oracle Corporation', 'QUALCOMM Incorporated', 'Intuit Inc.', 'Tesla, Inc.', ...	International Business Machines Corporation', 'QUALCOMM Incorporated', 'Adobe Inc.', 'Tesla, Inc.', ...
2019	'Facebook, Inc.', 'Applied Materials, Inc.', 'Tesla, Inc.', 'Microsoft Corporation', ...	'Alphabet Inc. (Class C)', 'Applied Materials, Inc.', 'Tesla, Inc.', 'Adobe Inc.', ...
2020	'Alphabet Inc. (Class C)', 'Applied Materials, Inc.', 'Tesla, Inc.', 'Adobe Inc.', ...	'Broadcom Inc.', 'Accenture plc', 'Tesla, Inc.', 'Verizon Communications Inc.', ...
2021	'NVIDIA Corporation', 'Accenture plc', 'Intuit Inc.', 'Applied Materials, Inc.', ...	'NVIDIA Corporation', 'Automatic Data Processing, Inc.', 'Adobe Inc.', 'Applied Materials, Inc.', ...
2022	'Broadcom Inc.', 'Automatic Data Processing, Inc.', 'Intuit Inc.', 'NVIDIA Corporation', Incorporated', ...	'Microsoft Corporation', 'Oracle Corporation', 'Adobe Inc.', 'NVIDIA Corporation', ...

B. Clustering for cryptocurrency selection

The volatility of cryptocurrency data using two evolving crypto indices has been analyzed. The beta measurement with the Crypto20 index was unclear, but the S&P CME bitcoin futures-based crypto index showed more sensible values, as seen in Table 4.7. This research has used k-means clustering to select assets and created two portfolios, C-01 and C-02, based on the highest and second-highest mean returns from each of the 3 crypto clusters. Table 4.8 captures a complete list of crypto assets for each portfolio.

Table 4.7: Beta Values of Cryptocurrencies

Cryptocurrency	Crypto20 index	S&P CME Bitcoin futures index	Cryptocurrency	Crypto20 index	S&P CME Bitcoin Futures Index
Bitcoin	0.000	0.871	File Coin	-0.001	0.972
Cardano	-0.001	0.975	Ethereum Classic	-0.003	0.975
Binance Usd	0.000	-0.004	Polygon	-0.002	1.102
Tether	0.000	0.002	Aave	-0.002	1.173
Solana	0.001	0.554	Tron	0.000	0.613
Algorand	0.000	0.801	Stellar	0.001	0.820
Terra Luna	0.005	0.836	Vchain	0.001	0.970
Shiba Inu	0.000	1.007	Elrond	0.001	0.815
Binance Coin	0.000	0.696	Litecoin	0.000	0.842
Avalanche	-0.002	1.109	Ethereum	-0.002	0.954
Chain Link	-0.003	0.933	Ripple	0.001	0.769
Serum	0.001	0.810	Dogecoin	0.001	0.827
Bitcoin Cash	-0.001	0.895			

Table 4.8: Crypto-based Portfolio Clusters

C-01 crypto-based portfolio	C-02 crypto-based portfolio
Ethereum Classic	Ripple
Terra Luna	Terra Luna
Tron	Tether

4.3.3 Portfolio Construction and Optimization

This research constructs two distinct types of portfolios. From the clusters of stocks, two portfolios (K-01 and K-02) were selected, and similarly, two portfolios (C-01 and C-02) were selected from cryptocurrencies. K-01 comprises stocks with the highest mean returns, while K-02 includes stocks with the second highest mean returns. Likewise, cryptocurrencies in C-01 and C-02 are chosen based on the same criteria.

Various traditional portfolio construction techniques are applied especially for the years 2020 and 2022. This study identifies the year 2022 as a risky year with many stocks performing with

negative mean returns. Moreover, for technological stocks, the year 2020 showcased positive returns.

This research applied traditional techniques such as equal weighted (EW), inverse variance weighted (IVW), mean-variance optimization (MVO), constrained portfolio optimization (CPO), and Sharpe ratio optimization (SRO) to optimize asset weights for the years 2020 and 2022. Subsequently, data-driven portfolio optimization techniques were utilized on the portfolios. A significant improvement in profitability was observed when diversification through clustering was employed. These results are discussed in Chapter 5.3.

4.4 Investigation of Clustering Algorithms for Diversification

Diversifying asset allocation is a crucial aspect of building a profitable portfolio. The resiliency of the portfolio depends on the optimization techniques as well as algorithms used in the asset allocation. Clustering techniques would help in designing a diversified portfolio. This study investigates the resiliency of different traditional and recently proposed data-driven portfolio techniques in conjunction with four clustering techniques under varying market conditions. The novelty of the study is to present a resilient portfolio optimization using DBSCAN and AP clustering techniques.

Previous experimentation on using clustering algorithms for diversification has indicated potential profitability; however, the study was confined to a single clustering algorithm. Additionally, apart from k-means, other clustering algorithms have not been extensively explored in financial applications like portfolio construction. In this experiment, various clustering methods in conjunction with different PO techniques are used. The findings propose efficient strategies for leveraging diverse clustering methods alongside various PO approaches to maximize profitability in volatile market conditions.

4.4.1 Dataset

This research considered stock data from the following sectors: Technology (Apple Inc. - AAPL, Microsoft Corporation - MSFT, Amazon.com Inc. - AMZN, Alphabet Inc. (Google) - GOOGL, Facebook Inc. - META), Consumer (The Coca-Cola Company - KO, McDonald's Corporation - MCD, Procter & Gamble Company - PG, Dollarama Inc. - DOL, Walmart Inc. - WMT), Energy (Exxon Mobil Corporation - XOM, Chevron Corporation - CVX, Canadian Natural Resources Limited - CNQ, Enbridge Inc. - ENB, Suncor Energy Inc. - SU), Automobiles (Tesla - TSLA, General Motors Company - GM, Toyota - TM, Honda Motor Co., Ltd. - HMC, BMW AG - BMWYY), Pharma (Johnson & Johnson - JNJ, Pfizer Inc. - PFE, Merck & Co., Inc. - MRK, Novartis AG - NVS, AstraZeneca PLC - AZN), Supply Chain (United Parcel Service, Inc. - UPS, FedEx Corporation - FDX, J.B. Hunt Transport Services Inc. - JBHT, ArcBest Corp. - ARCB, Forward Air Corp. - FWRD), and Banking Sector (Royal Bank of Canada - RY, Toronto-Dominion Bank - TD, Bank of Nova Scotia (Scotiabank) - BNS, Bank of Montreal - BMO, Canadian Imperial Bank of Commerce - CM). A total of 35 stocks, 5 from each of the seven sectors, have been selected. The data considered spans from 2019 to 2022. Each year has been individually explored and experimented within this study.

4.4.2 Implementation

A. Exploratory analysis

The preliminary analysis indicates that the year 2020 exhibited substantial profitability, contrasting with the year 2022 wherein a significant proportion of stocks yielded negative mean returns. For example TESLA (TSLA), throughout the year 2020 showed an upward trend but in 2022, it dropped continuously. Table 4.9 summarizes the cluster number for each year using different clustering algorithms. Different clustering algorithms function differently resulting in varying cluster numbers for different years. For each year this study has considered two different portfolios. From each cluster, the stock with the highest mean return has been selected to be included in the **K-01** portfolio, whereas the stock with the second highest mean return is

	kmeans	hierarchical	DBSCAN	AP
2019	12	12	13	9
2020	12	10	14	6
2021	12	12	15	8
2022	12	8	15	8

Table 4.9: Number of clusters using different clustering techniques

included in the **K-02** portfolio.

B. Clustering for stock selection

In this investigation, the covariance matrix of stock returns is computed for each year. This covariance matrix represents the relationships between the stocks for each respective year. The covariance matrix can be denoted as:

$$\Sigma = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X}) \quad (4.1)$$

- Σ is the covariance matrix.
- n is the number of observations (rows in the data).
- X is the data matrix where each row represents an observation and each column represents a variable.
- $\bar{X}$ is the matrix of mean values, where each column contains the mean of the corresponding variable in X .
- $(X - \bar{X})$ is the matrix of deviations of each observation from the mean.
- $(X - \bar{X})^T$ is the transpose of the matrix of deviations.

The covariance matrix is then utilized to determine the optimal number of clusters for both k-means and hierarchical clustering, employing the Within-Cluster Sum of Squares (WCSS) criterion. In hierarchical clustering, dendrograms for each year were meticulously examined

to ascertain the appropriate cluster count. For DBSCAN, a reachability⁴ plot was employed annually to identify distinct clusters.

The AP algorithm autonomously determines the number of clusters in the annual dataset. In this study, the covariance matrix derived from stock returns was also employed in AP clustering.

4.4.3 Portfolio Construction and Optimization

From each cluster generated by the aforementioned clustering algorithms, this study formed two portfolios: K-01 and K-02. For K-01, stocks with the highest mean returns from each cluster were selected, while for K-02, stocks with the second highest mean returns were chosen. This is done for each clustering approach and then both traditional and data-driven PO techniques are applied for further experimentation.

Following portfolio construction, weight optimization is performed using both traditional and data-driven algorithms. This research utilizes several notable traditional algorithms including EW, IVW, MVO, CPO, and SRO.

Various PO techniques exhibit distinct behaviors across different clustering approaches. This study identified suitable clustering technique(s) to maximize profitability. The investigation applied clustering technique(s) and demonstrated their effectiveness in achieving maximum profitability, particularly under volatile market conditions. These results are discussed in Chapter 5.4.

⁴A reachability plot is a visual tool in DBSCAN clustering, depicting each point's reachability distance as processed by the algorithm. This distance signifies how far a point is from its nearest core point. It displays the reachability distance of each point in the order processed by DBSCAN, where it quantifies how far a point is from its nearest core point that can be reached within a specified minimum distance and the minimum number of points. To find the number of clusters, valleys, or dips in the reachability plot indicate cluster boundaries.

Chapter 5

Results, Evaluation, and Discussion

This chapter presents the results of the experiments detailed in Chapter 4. The previous chapter elaborated on the five experimental setup and algorithmic strategies. This chapter analyzes the outcomes and demonstrates how these strategies improve the predictive accuracy of financial assets and profitability in PO.

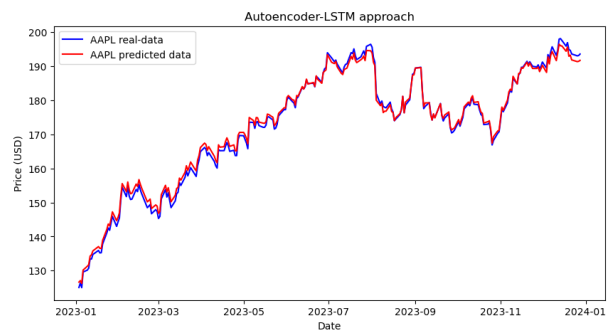
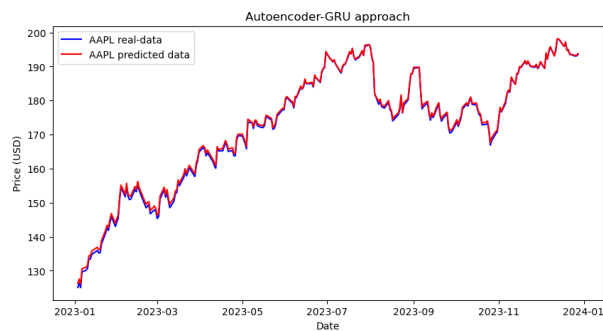
5.1 Encoder-Decoder Architecture for Financial Price Prediction

This study introduces two novel encoder-decoder architectures based on LSTM and GRU, for financial price prediction (see Chapter 4.1). These models were tested on stocks, indices, and cryptocurrencies. The results are detailed below.

5.1.1 Predictions for Individual Stock

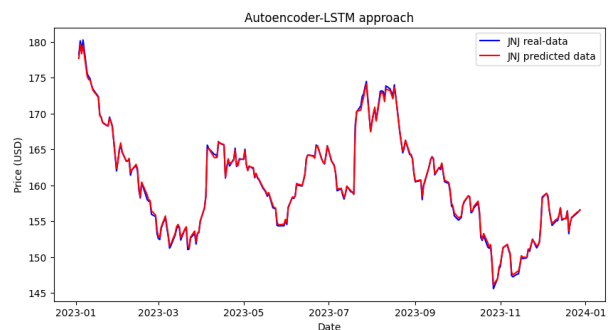
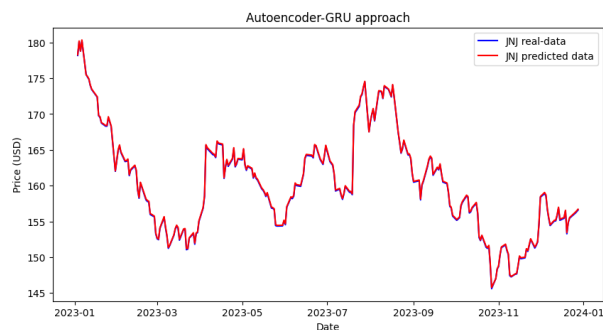
The extensive experimentation encompassed three activation functions and adjustments to hyperparameters, including 1st-moment exponential decay rate β_1 , 2nd-moment exponential decay rate β_2 , learning rate, and numerical stability set at 1×10^{-7} . Among various configurations of AE-LSTM, the setup with 1st-moment exponential decay rate, β_1 at 0.9, 2nd-moment exponential decay rate β_2 at 0.999, learning rate as 0.001, numerical stability at 1×10^{-7} , and

ELU as the activation function proves to be the most effective. However, the proposed AE-GRU architecture with ReLU as the activation function consistently outperforms AE-LSTM in these assessments for AAPL, JNJ, JPM, and CVX.



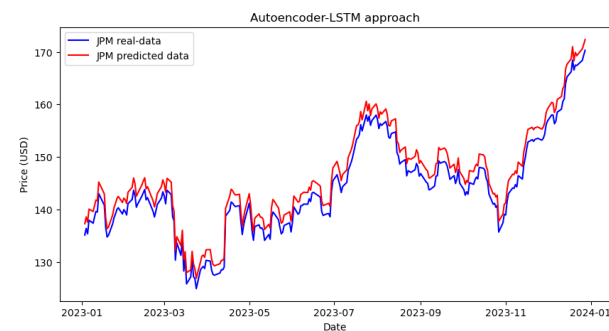
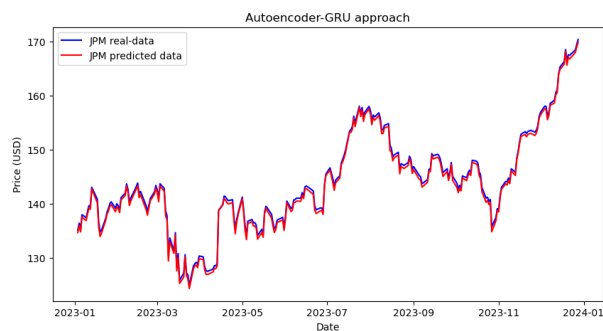
AE-GRU for AAPL

AE-LSTM for AAPL



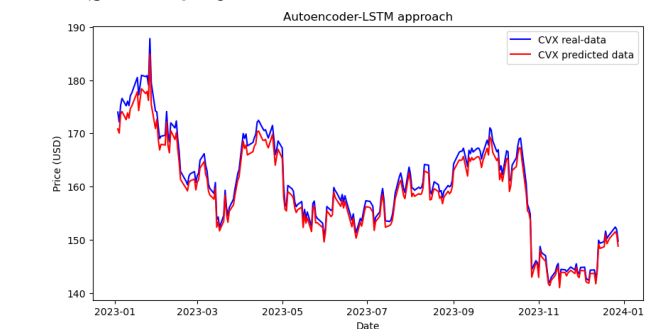
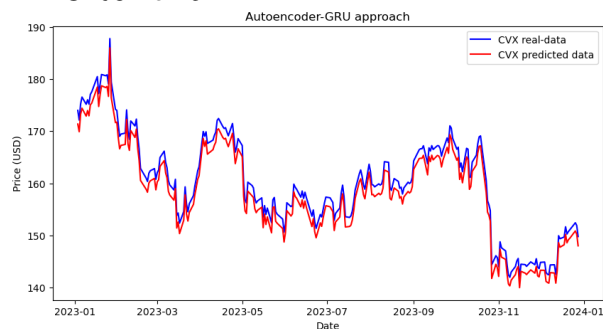
AE-GRU for JNJ

AE-LSTM for JNJ



AE-GRU for JPM

AE-LSTM for JPM



AE-GRU for CVX

AE-LSTM for CVX

Figure 5.1: Trend prediction by AE-GRU and AE-LSTM for individual stocks.

Table 5.1: Evaluation of the performance of AE-LSTM and AE-GRU through metrics.

Components	Metric	Algorithms	
		AE-GRU	AE-LSTM
AAPL	MSE	0.37	1.22
	RMSE	0.61	1.10
	MAE	0.53	0.97
	R-Squared	0.999	0.996
	MAPE	0.3%	0.6%
JNJ	MSE	0.01	0.07
	RMSE	0.12	0.26
	MAE	0.1	0.2
	R-Squared	0.9997	0.9987
	MAPE	0.07%	0.12%
JPM	MSE	0.33	4.79
	RMSE	0.58	2.18
	MAE	0.55	2.17
	R-Squared	0.996	0.94
	MAPE	0.4%	1.51%

Figure 5.1 visually demonstrates the superiority in providing an accurate prediction of stock trends by AE-GRU compared to AE-LSTM-based predictions.

Table 5.1 presents performance metrics for the proposed novel architectures, AE-GRU and AE-LSTM, across various components such as AAPL, JNJ, and JPM. The metrics evaluated include MSE, RMSE, MAE, R-squared coefficient, and MAPE. Table 5.1 demonstrates the superiority of AE-GRU over AE-LSTM is affirmed by these five evaluation metrics.

For the AAPL stock:

- AE-GRU exhibits lower MSE, RMSE, MAE, and MAPE values compared to AE-LSTM, indicating superior performance in predicting AAPL's closing prices.
- Additionally, AE-GRU achieves higher R-Squared values, signifying better goodness-of-fit between predicted and actual values for AAPL.

Similarly, for JNJ and JPM stocks:

- AE-GRU consistently outperforms AE-LSTM across all metrics, demonstrating its efficacy in predicting closing prices for both JNJ and JPM.
- Notably, AE-GRU achieves higher R-Squared values, indicating stronger linear relationships between predicted and actual values for JNJ and JPM.

Despite the similar heightened volatility as observed in Table 4.1 and strong correlations between technology and banking stocks, AAPL and JPM exhibited observable variations in the evaluation metrics, as presented in Table 5.1. This discrepancy arises due to differences in the rates of volatility change between AAPL and JPM, as depicted in Figure 4.2 (a) and (c). While AAPL demonstrates a low magnitude of volatility change, it experiences a rapid increase. Conversely, JPM showcases a low magnitude of change with minimal fluctuations in volatility. Varying rates of volatility disrupt the performance of the AE-LSTM model to a certain extent but the AE-GRU model shows robustness in such cases.

Thus, for general stocks of different categories, such as technology, banking, healthcare, and energy, it can be concluded that the proposed models perform with higher accuracy. In every scenario, AE-GRU outperforms the AE-LSTM.

Such performance efficacy cannot be observed in any baseline models such as vanilla LSTM and GRU [92] [93].

5.1.2 Predictions for the Stock Index- S&P

Hyperparameter tuning for AE-GRU involves adjusting key parameters, such as setting 1st-moment and 2nd-moment exponential decay rates β_1 and β_2 to 0.9 and 0.999 respectively, the learning rate to 0.001, and maintaining numerical stability at 1×10^{-7} . Specifically for stock indices like S&P, AE-GRU performs optimally when using `tanh` as the activation function. In contrast, AE-LSTM with a `ReLU` activation function exhibits favorable outcomes compared to its other variants.

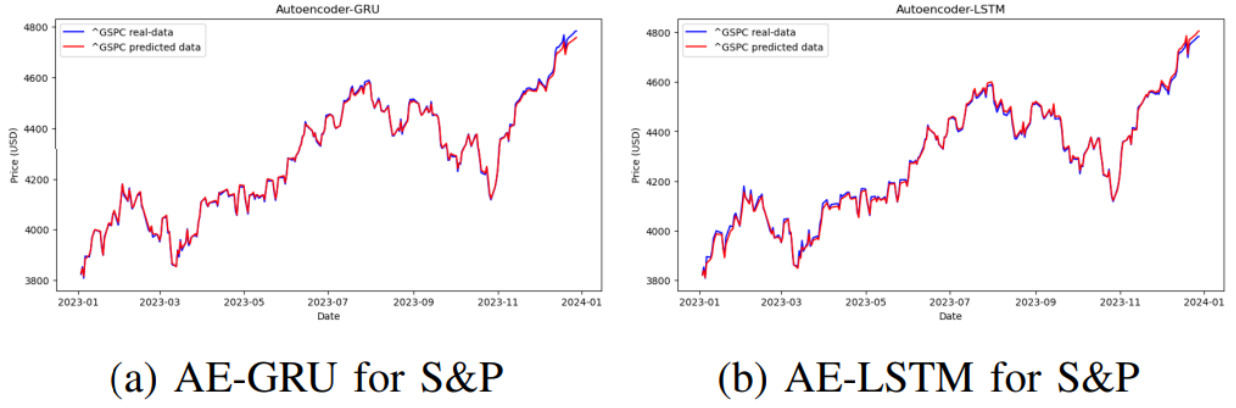


Figure 5.2: Trend prediction by AE-GRU and AE-LSTM for stock index.

However, in direct comparison, the `tanh`-based AE-GRU outperforms all variations of AE-LSTM in forecasting annual prices of S&P, as depicted in Figure 5.2. Despite a slightly higher MSE attributed to the large scale of values, the proposed AE-GRU architecture significantly outperforms AE-LSTM with an impressively low MAPE of 0.1%. Consequently, for stock index prediction, the proposed AE-GRU-based architecture demonstrates superiority over AE-LSTM which is illustrated in Figure 5.2.

5.1.3 Predictions for the Cryptocurrency- Bitcoin

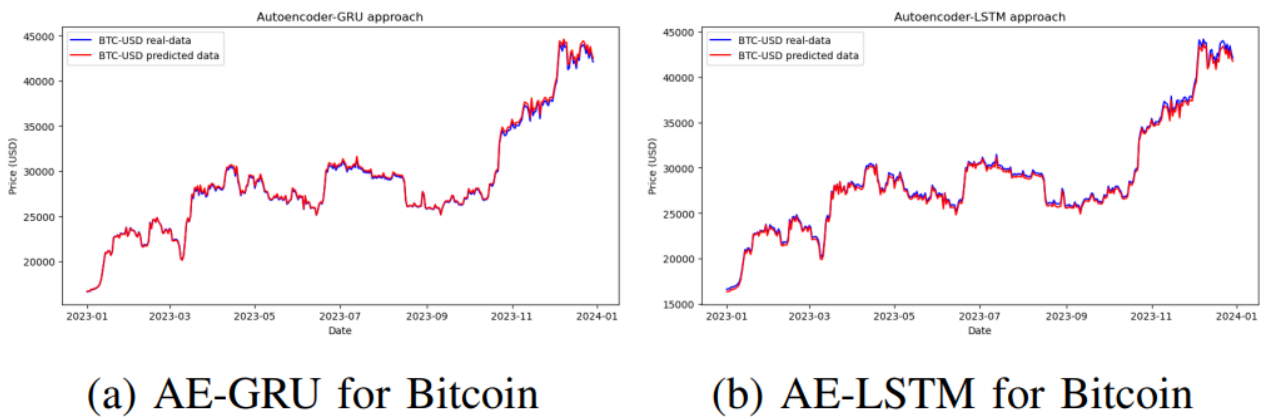


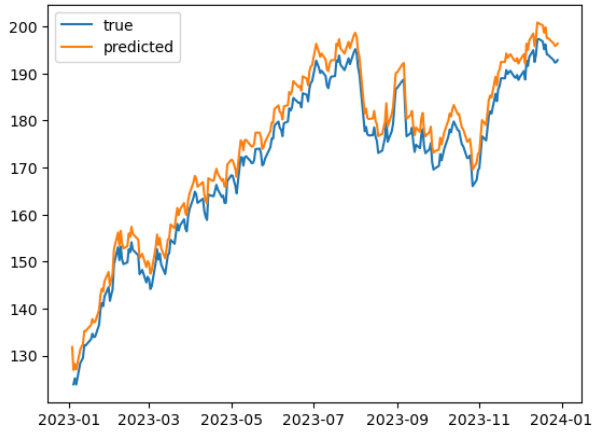
Figure 5.3: Trend prediction by AE-GRU and AE-LSTM for cryptocurrency.

Figure 5.3 demonstrates that the proposed AE-GRU design, utilizing ReLU as the activation function, outperforms the AE-LSTM prediction. Although hyperparameter tuning, employing

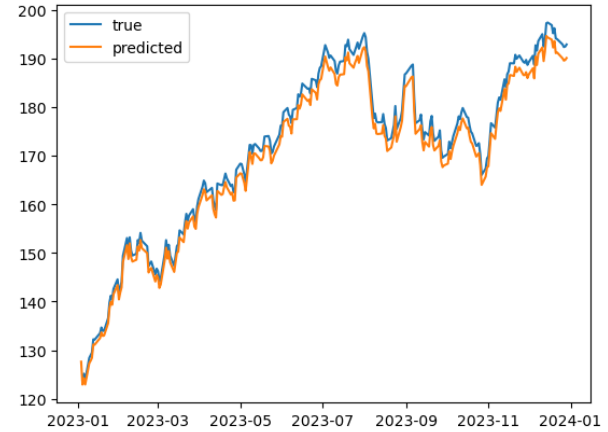
ELU as the activation function, enhanced the predictive accuracy of the AE-LSTM structure to some degree, it still fell short of surpassing the novel AE-GRU design. The proposed AE-GRU architecture exhibited superior performance to the AE-LSTM architecture, achieving a MAPE of 0.5%, which is half the value observed in AE-LSTM.

5.1.4 Assessing the Efficacy against the Baseline LSTM and GRU Models

To the best of our knowledge, this approach's predictive performance improvement is unmatched by any existing baseline (LSTM and GRU) algorithms. Numerous studies on one-day ahead price prediction have been reviewed, and none have demonstrated such effectiveness, particularly with baseline LSTM and GRU models.



(a) Performance of LSTM for AAPL stock prediction



(b) Performance of GRU for AAPL stock prediction

Figure 5.4: Trend prediction by LSTM and GRU for AAPL stock.

Figure 5.4 depicts the 2023 price prediction for AAPL stock. The illustration reveals that the proposed encoder-decoder-based variations perform better than the baseline LSTM and GRU algorithms.

Stocks	Evaluation Metrics	LSTM	GRU
AAPL	MSE	11.80	4.38
	RMSE	3.44	2.09
	MAE	3.43	2.02
	MAPE	12.06	11.59
	R-squared	0.96	0.98

Table 5.2: Evaluation Metrics for AAPL using LSTM and GRU models

Table 5.2 presents the performance of LSTM and GRU models in predicting AAPL stock prices. The proposed encoder-decoder-based LSTM and GRU models (refer to Table 5.1) show better predictive accuracy than the baseline LSTM and GRU models.

5.1.5 Practical Applications

Cakici et al. [94] revisited evidence on market risk premium predictability by equity anomalies, extending findings internationally and across different anomaly sets, finding that anomalies fail to predict market excess returns due to methodological choices impacting prediction performance. Similarly, our research could be extended in this direction to predict different economic factors such as risk premiums by incorporating features such as market anomaly. The novel encoder-decoder models (AE-GRU and AE-LSTM) developed in this study incorporating LSTM and GRU layers, AE-LSTM, and AE-GRU, demonstrate efficacy in predicting various categories of stocks, cryptocurrencies, and stock indices. These models can be an important tool to assist traders and investors in making informed decisions regarding their investment portfolios. By leveraging the model's predictions, traders and investors can make informed decisions regarding their investment portfolios. Furthermore, the insights provided by the model can be utilized to formulate trading strategies based on simple indicators such as shape ratios and other technical analysis tools. These strategies can help traders identify optimal entry and exit points, manage risk more effectively, and potentially enhance their overall trading performance. Overall, the model's predictions have the potential to contribute to more informed and data-driven decision-making in financial markets, leading to improved trading outcomes. Moreover, considering transaction costs is crucial in assessing the economic gains derived from

trading strategies based on predictive models. Integrating transaction costs into the evaluation of trading strategies is essential for ensuring their viability in real-world trading environments. The inclusion of features in the ML models increases learning capacity. Thus, adding transaction costs as a new feature might enhance the model's performance positively in practical scenarios allowing an increase in the efficacy of the model in predicting buy and sell signals more robustly. The addition of such features in the model in such scenarios requires careful experimentation which will be a natural future extension of this research.

5.1.6 Key Findings

The proposed architecture, AE-GRU, offers an innovative approach by combining encoder-decoder architecture and GRU, demonstrating superior performance compared to the proposed AE-LSTM across various market scenarios. This research has shown experiments using stocks from different sectors and also included- S&P as the stock index, and bitcoin as a cryptocurrency. Across all sectors, the proposed novel encoder-decoder architecture, AE-GRU, consistently outperformed the AE-LSTM. A novel AE-LSTM architecture is also designed in this study, which improvised the existing idea of LSTM and autoencoder combination through rigorous hyperparameter tuning. Despite experimenting with hyperparameter tuning for AE-LSTM and improvising its performance from the existing version, it still could not match the prediction accuracy of the proposed novel AE-GRU architecture. Experiments were organized on three different computing devices, yielding similar results with slight variations in the values of evaluation metrics due to randomized weight initialization. Experiments have considered training data from significant market events of the 2001 high-tech bubble burst, the 2008 global financial crisis, and the 2020 COVID-19 stock market crisis. Our findings advocate for the adoption of the novel AE-GRU architecture, showcasing enhanced predictive accuracy for financial assets across various sectors and market conditions. The proposed encoder-decoder-based GRU architecture retains the long-term non-linear contexts from the past temporal observations resulting in superior efficacy in predicting future values for stocks, indexes, and highly volatile cryptocurrencies. After significant enhancements were made to the AE-LSTM architecture, the

newly proposed AE-GRU architecture demonstrated superior performance. This research proposes two novel encoder-decoder architectures, AE-LSTM and AE-GRU that provides higher predictive accuracy than the existing models.

5.2 Hybrid Data-Driven DL-based PO

This study proposes a novel DL-based PO algorithm that fuses two RNN algorithms with a data-driven PO approach (see Chapter 4.2). This method combines improved predictive accuracy, resiliency through diversified asset selection using clustering, and a statistically significant data-driven PO strategy. The experiments also assess the efficacy of bidirectional algorithms and offer strategies for effective fusion in DL-based PO design. The experimental results are detailed below.

5.2.1 Performance Analysis of BiLSTM and BiGRU

This experiment employed 2023 as the testing period. The DL models were trained on data up to 2022, and these trained models were then utilized to predict prices for 2023.

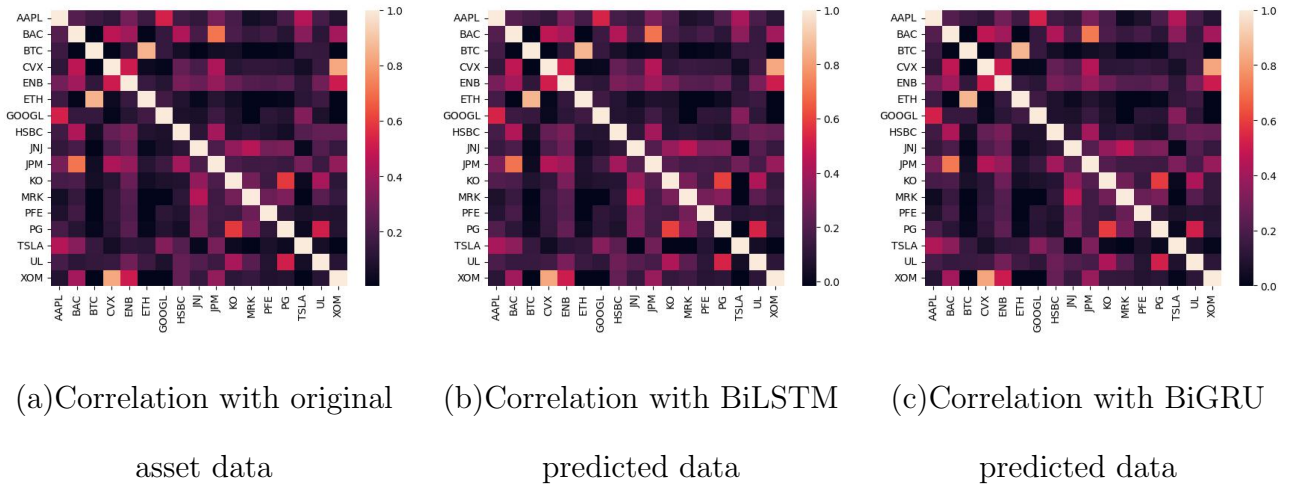


Figure 5.5: Correlation analysis using actual data and forecasted data of 2023

The correlation among various financial assets analyzed in this study for the year 2023 is depicted as a heat map in Figure 5.5. The BiLSTM and BiGRU DL models were trained to

forecast asset prices for the same year. In this examination, the correlation between assets is assessed using both the predicted data from the models and the actual data.

The correlation matrix derived from the actual data of 2023 Figure 5.5(a) reveals a robust correlation between stocks in the energy sector and those in the banking sector, particularly Bank of America (BAC). Additionally, BTC demonstrates a significant association with ETH. Furthermore, stocks within the same sector consistently exhibit elevated levels of correlation.

The correlation analysis in the Figure 5.5(b & c) indicates that the predicted data successfully incorporates all the significant correlations observed among financial assets in the actual data. This suggests that the DL models have effectively captured the correlations among assets essential for constructing portfolios.

Furthermore, upon reviewing Table 5.3, it becomes evident that BiGRU exhibits superiority over BiLSTM in forecasting the yearly prices of financial assets. The four evaluation metrics examined in this research consistently endorse the enhanced performance of BiGRU compared to BiLSTM in the prediction of financial asset prices.

Figure 5.6 presents the performance of BiGRU and BiLSTM models in various financial sectors. Notably, the predictions generated by the BiGRU model exhibit greater efficiency compared to those produced by the BiLSTM model. As illustrated in Table 5.3, a comparative analysis reveals the superior performance of BiGRU-based predictions over BiLSTM for both AAPL and the highly volatile BTC-USD. Specifically, for AAPL, the evaluation metrics indicate a lower MSE of 0.14 for BiGRU in contrast to 0.40 for BiLSTM, signifying the enhanced accuracy of BiGRU predictions. This suggests that the predicted values generated by BiGRU closely align with the actual values. Moreover, BiGRU exhibits a reduced MAE compared to BiLSTM, indicating a lower error in BiGRU predictions. The error percentage for BiGRU is notably lower at 0.18%, while for BiLSTM, it stands at 0.30% in the case of AAPL. This observation is evident in Figure 5.6(a & b), where the predictions for AAPL using BiGRU are notably sharp and accurate in contrast to those based on BiLSTM. In the context of highly volatile cryptocurrencies, such as BTC-USD, BiGRU outperforms BiLSTM, demonstrating superior performance with a minimal MSE of 0.0003. Moreover, when considering MAE, BiGRU

Table 5.3: Performance Comparison of BiLSTM and BiGRU

Data Series	Metric	Algorithms	
		BiLSTM	BiGRU
AAPL	MSE	0.40	0.14
	MAE	0.54	0.30
	R-Squared	0.998	0.999
	MAPE	0.30%	0.18%
BAC	MSE	0.08	0.03
	MAE	0.27	0.17
	R-Squared	0.9901	0.9959
	MAPE	0.90%	0.58%
BTC-USD	MSE	0.0006	0.0003
	MAE	0.019	0.014
	R-Squared	0.9994	0.9997
	MAPE	14.7%	11.1%
CVX	MSE	0.63	0.11
	MAE	0.78	0.25
	R-Squared	0.993	0.999
	MAPE	0.49%	0.16%
JNJ	MSE	0.2	0.2
	MAE	0.4	0.4
	R-Squared	0.99	0.99
	MAPE	0.3%	0.3%
PG	MSE	0.5	0.3
	MAE	0.7	0.5
	R-Squared	0.98	0.99
	MAPE	0.5%	0.3%

exhibits lower values. In terms of error percentage, predictions based on BiGRU exhibit a 0.58% error, whereas BiLSTM-based predictions show a higher error rate of 0.90%. This observation is evident in Figure 5.6(k & l), where the predictions for BTC generated by BiGRU are characterized by sharp precision, contrasting with the less precise predictions produced by BiLSTM.

Similar observations can be made for the financial assets of the other sectors as well. Therefore, while both BiLSTM and BiGRU demonstrate accurate price predictions and closely align

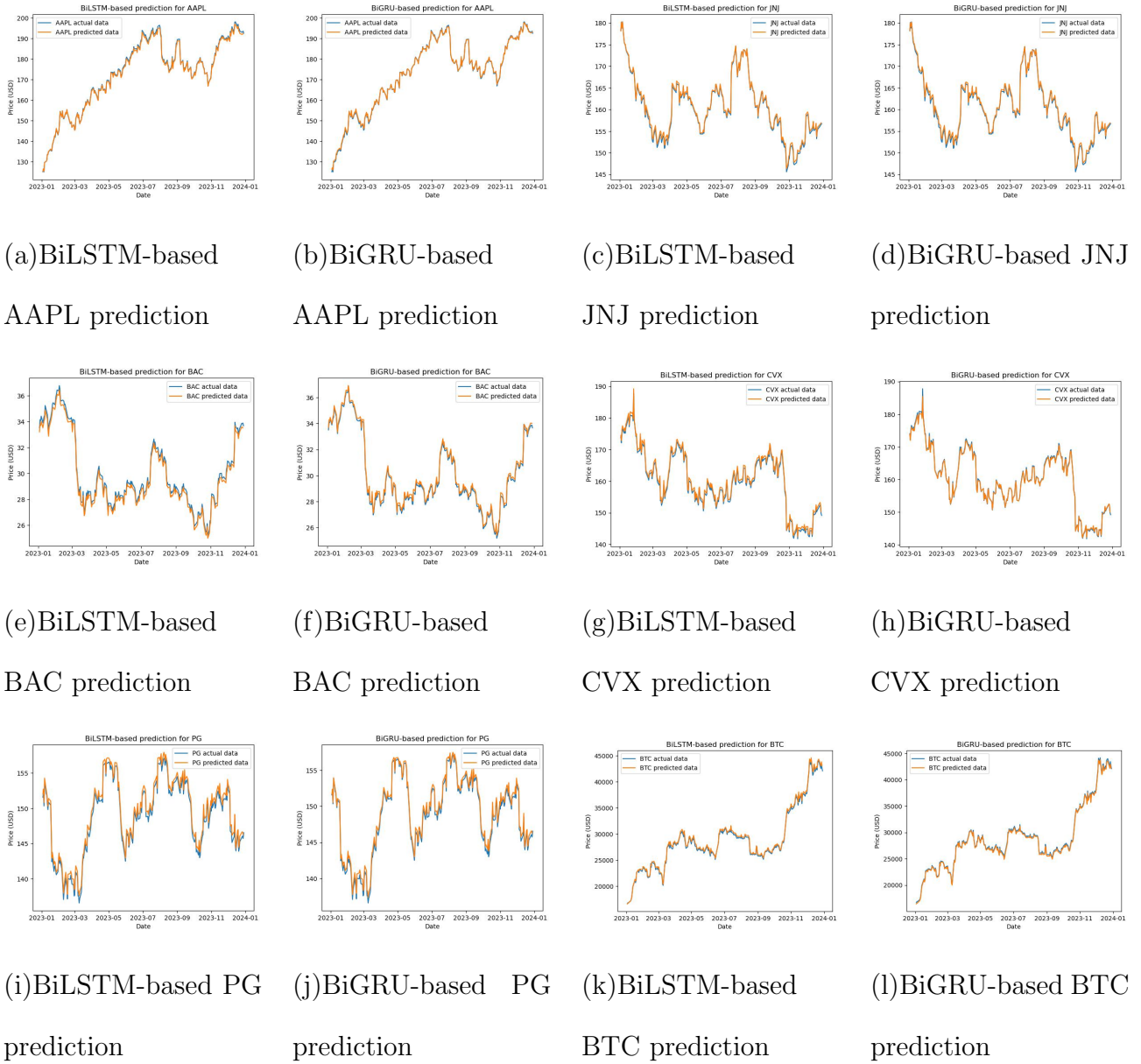


Figure 5.6: BiLSTM and BiGRU predictions

with the actual data, the BiGRU algorithm outperforms the BiLSTM algorithm, particularly for highly unpredictable assets, showcasing greater accuracy.

5.2.2 Portfolio Performance

In the creation of financial portfolios, this study explores the Data-Driven approach, taking into account essential features of financial time series, such as non-normality, which are often overlooked by traditional methods. These features play a crucial role in constructing robust portfolios. Additionally, the study incorporates the concept of diversification in stock selec-

tion. The performance of portfolios is evaluated under two scenarios: (i) where all stocks are included without specific selection (no diversification), and (ii) where the AP clustering technique is employed to introduce diversity in stock selection. Consequently, the study assesses the performance of data-driven portfolios with and without clustering, providing insights into the impact of diversification on portfolio performance.

Without Diversification

This research has constructed portfolios encompassing 15 stocks and 2 cryptocurrencies, incorporating both real and forecasted data.

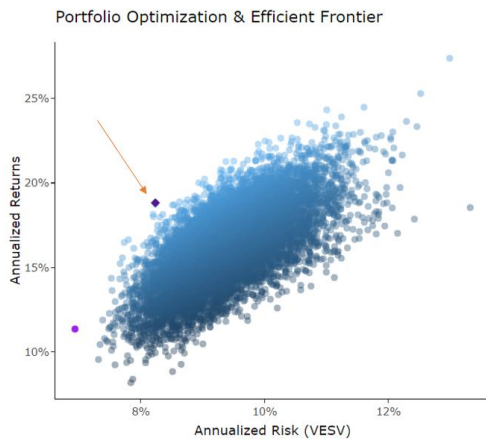
The efficient frontiers of the constructed portfolios are alike for both predicted and actual data, given the close resemblance between the actual values and the predicted values generated by both BiLSTM and BiGRU. The efficient frontiers constructed with four data-driven risk measures are illustrated in Figure 5.7. This figure presents the efficient frontiers for portfolios formed with four data-driven risk measures. In Figure 5.7 (b), it can be observed that the portfolio constructed with VES-based risk measure showed maximum profitability.

Figure 5.8 illustrates the optimization of portfolio weights under two paradigms: minimum risk and maximum Sharpe ratio for both actual and predicted values. Figure 5.8 (b) and (d) indicate a striking resemblance between portfolio weight optimization based on predicted price data and that based on actual data.

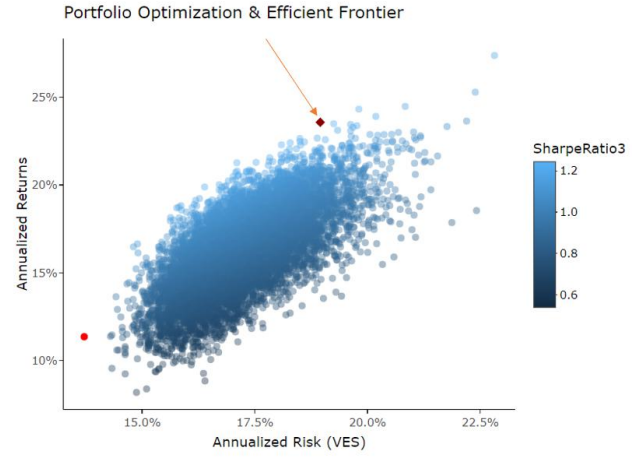
With Diversification through AP

Utilizing AP for stock selection revealed the presence of six clusters. To enhance diversification in the stock selection process, this research identifies the financial asset with the highest mean returns within each cluster. Consequently, the stocks chosen from each cluster are incorporated to construct a robust portfolio. This approach ensures diversification in the portfolio, a critical factor in bolstering its resilience.

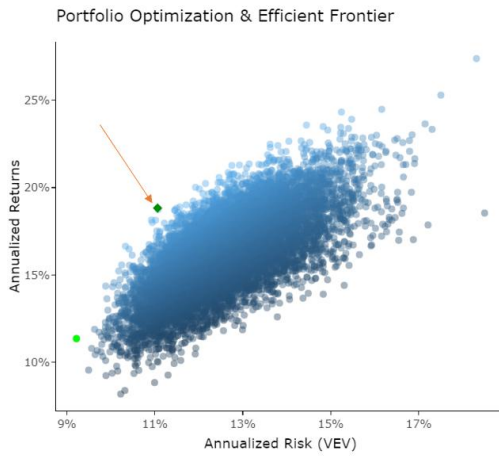
Figure 5.9 presents the performance of portfolios constructed with diversification under four data-driven risk measures. Here, portfolios constructed with both actual and predicted data



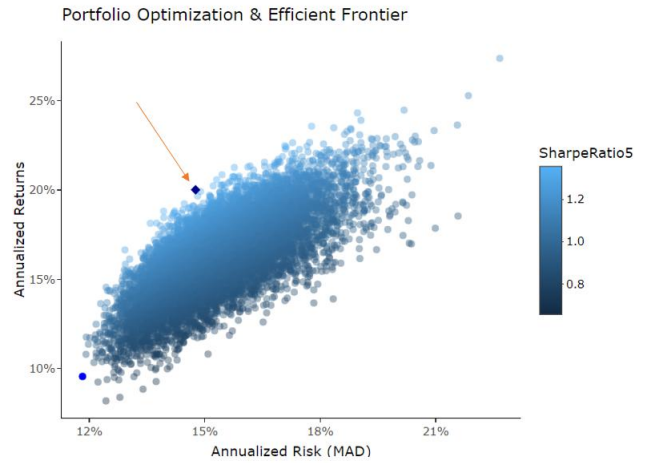
(a) Efficient Frontier based on VESV risk measure



(b) Efficient Frontier based on VES risk measure



(c) Efficient Frontier based on VEV risk measure



(d) Efficient Frontier based on MAD risk measure

Figure 5.7: Performance of portfolio based on data-driven risk measures

showed similar efficient frontiers. In the Figure 5.9(c), it could be observed that the portfolio constructed with VEV-based risk measure showed maximum profitability compared to the risk.

Upon comparing Figures 5.7 and 5.9, it is observed that the portfolios' performance doubled across all data-driven risk measures following the implementation of diversification through AP clustering for stock selection.

Figure 5.10 depicts the optimization of portfolio weights employing two approaches: minimizing risk and maximizing the Sharpe ratio for both actual and predicted values. A notable

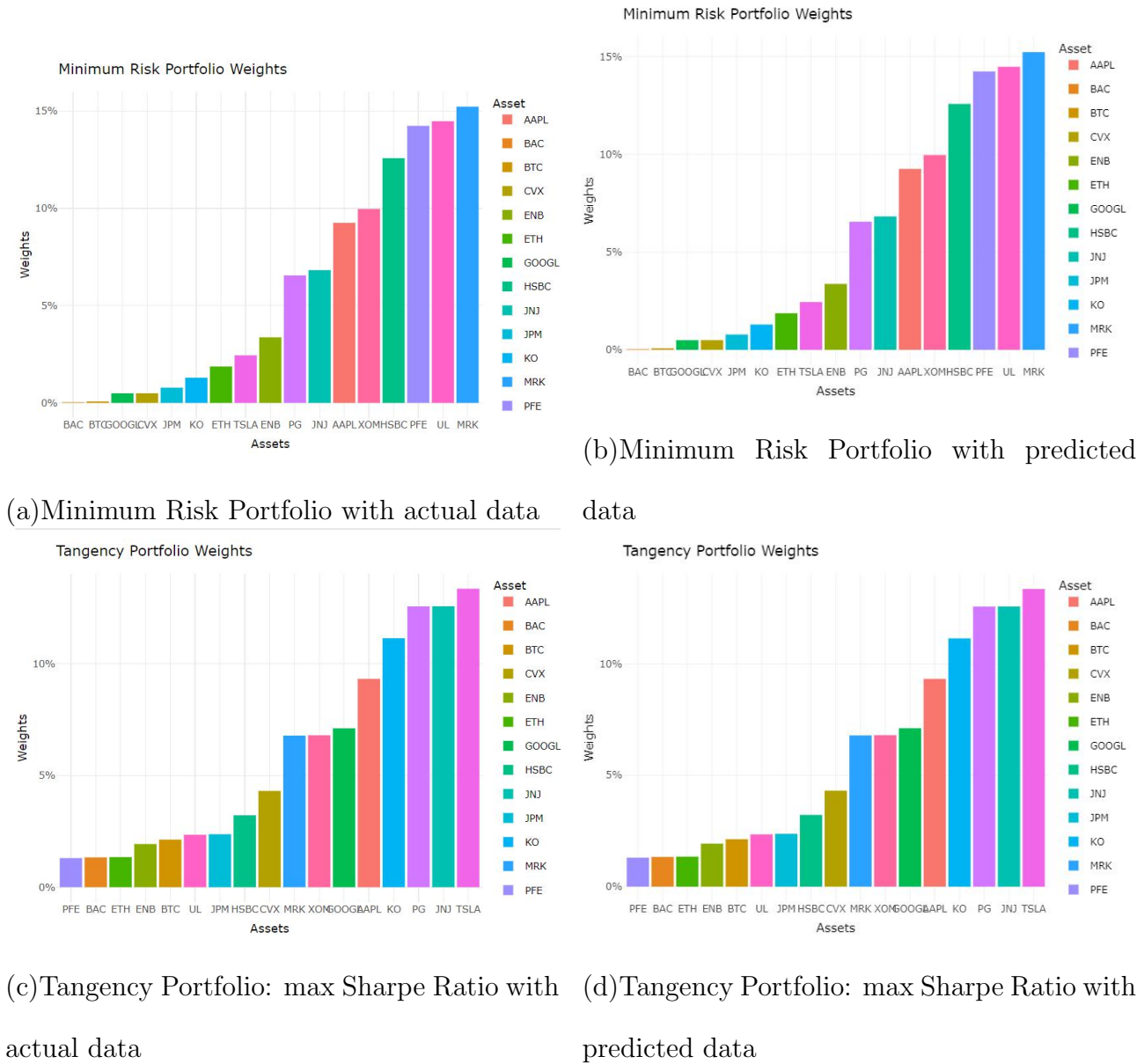


Figure 5.8: Portfolio Weight Distribution based on risk and Sharpe ratio: actual and predicted data

similarity in portfolio weight optimization between predicted and actual data is observed in Figure 5.10 (b) and (d).

The proficient performance of the BiLSTM and BiGRU algorithms in predicting financial asset prices across diverse market conditions leads to negligible discrepancies between actual and forecasted prices. Consequently, the data-driven algorithm exhibits comparable efficiency in portfolio weight optimization using both actual and data-driven approaches. The evaluation of these two algorithms suggests that the innovative integration of BiLSTM/BiGRU with the data-driven portfolio optimization algorithm constitutes a profitable technique. This approach

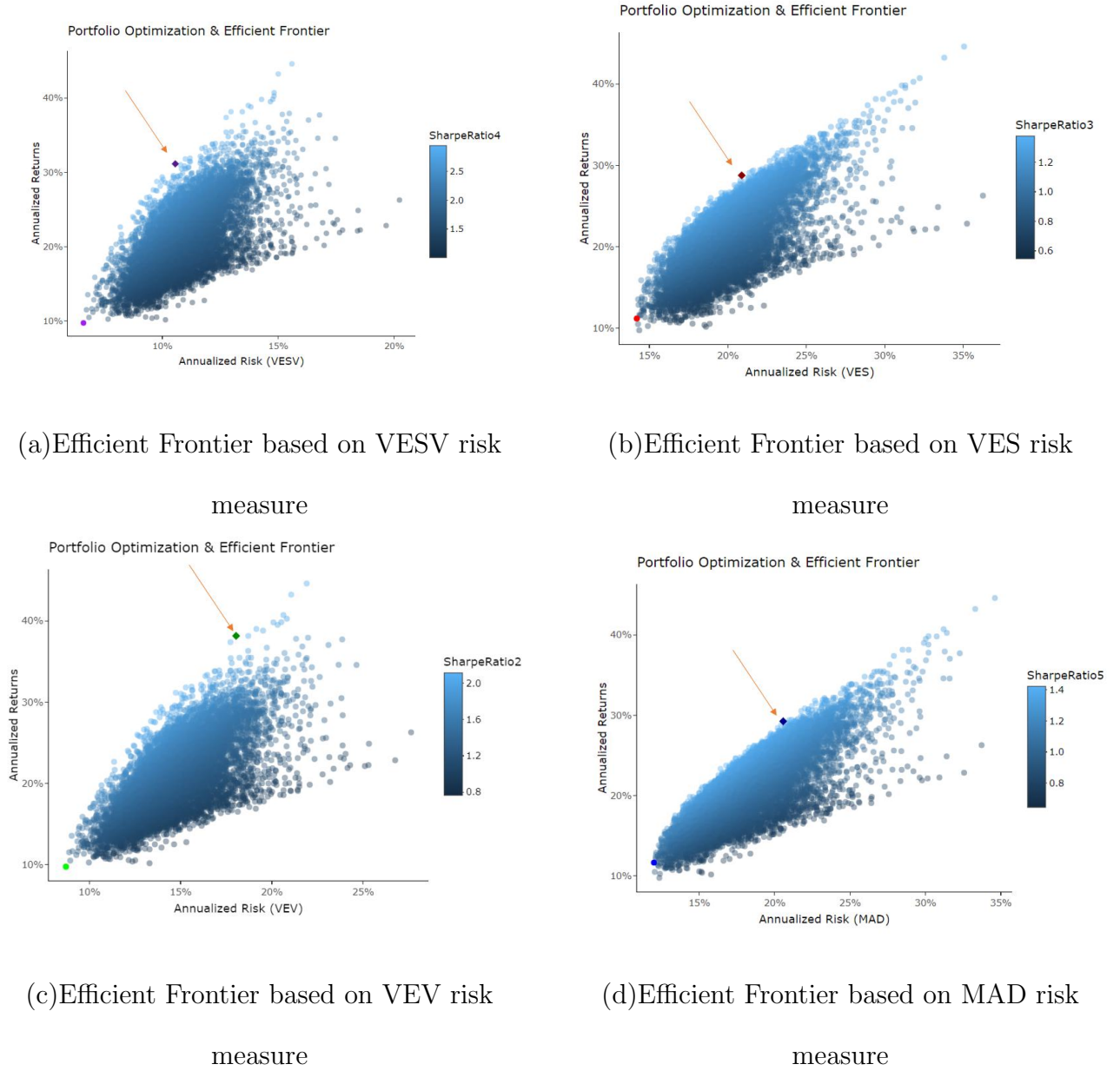


Figure 5.9: Performance of portfolio based on data-driven risk measures

demonstrates maximum profitability under diverse market conditions and with stocks from various sectors. In certain instances, in the domain of time series predictions, it has been observed that the fundamental GRU architecture outperforms the LSTM model, as documented by Yamak et al. [95]. Drawing a parallel from this observation, a similar trend is discerned in the comparison between BiGRU and BiLSTM architectures. The rationale behind this superiority of GRU-based architectures may be attributed to their inherent design characteristics, such as a simplified structure with fewer parameters and the absence of an explicit memory cell. These attributes enable GRU models to capture and retain relevant information more efficiently in

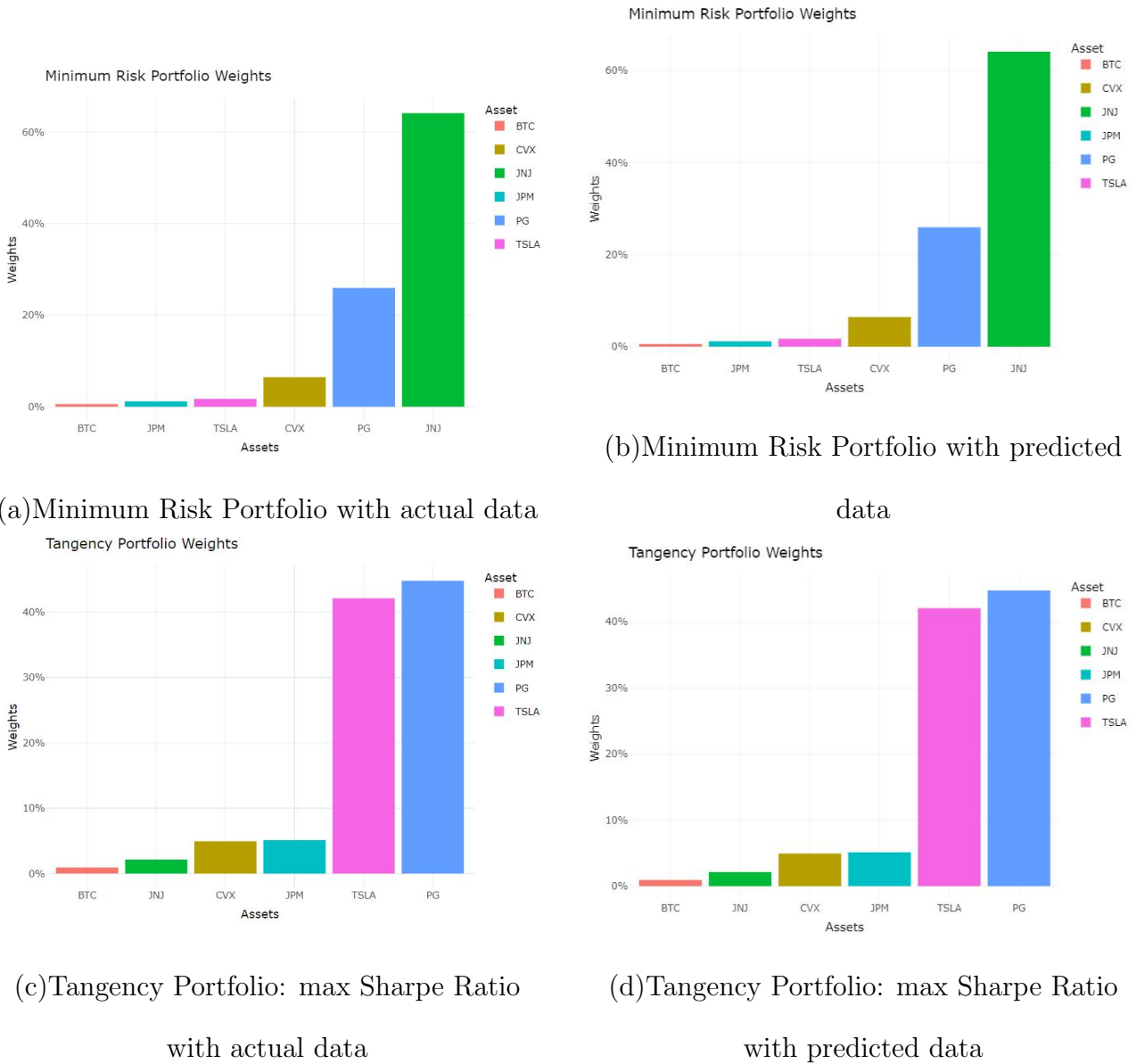


Figure 5.10: Portfolio Weight Distribution based on risk and Sharpe ratio with actual and predicted data

certain contexts, potentially leading to enhanced performance in specific time series prediction scenarios.

5.2.3 Key Findings

From the above discussions, key findings are summarized as follow:

1. Based on the observed results from the Table 5.3 and Figure 5.6, superior performance of BiGRU-based predictions is observed compared to BiLSTM predictions.
2. Comparing Figures 5.7 and 5.9, it is observed introduction of diversification through AP

- clustering algorithm enhances portfolio performance by twofold compared to portfolios without diversification.
3. In Figure 5.7(b), where portfolio construction lacks diversification, portfolios utilizing the VES-based risk measure exhibit superior performance compared to those constructed with alternative data-driven risk measures. Conversely, in Figure 5.9(c), with diversification incorporated into portfolio construction, portfolios employing the VEV-based risk measure demonstrate outperformance over portfolios constructed with other data-driven risk measures.
 4. Utilizing a data-driven approach for portfolio weight optimization, coupled with diversification through clustering techniques proves to be effective when implementing BiLSTM and BiGRU-based predictions.
 5. BiGRU demonstrates significant effectiveness in predicting prices compared to BiLSTM. Therefore, it is recommended to prefer BiGRU, particularly when dealing with dynamic market conditions and highly volatile assets, when constructing robust portfolios.

5.3 Resiliency in PO with Clustering Technique

This study primarily aims to enhance the resilience of traditional and data-driven portfolio optimization under high-risk market conditions by incorporating diversification through clustering technique (see Chapter 4.3). Experiments were done on two types of portfolios: stock-based and cryptocurrency-based. Additionally, the effectiveness of two lesser-known crypto indices was also investigated. The results from the experiments are discussed below.

5.3.1 Stock-based PO

Tables 5.4 and 5.5 display the performance of EW and IVW portfolios for each year's data and all years grouped.

However, these traditional portfolio construction techniques did not yield positive returns

Table 5.4: Summary Results for Equal Weight Portfolio

Dataset	Annualized Risk		Annualized Return		Diversification Index	
	K-01	K-02	K-01	K-02	K-01	K-02
2019-2022	30.91%	29.61%	26.78%	22.57%	1.34	1.36
2019	20.77%	20.77%	45.81%	39.40%	1.51	1.54
2020	40.10%	38.98%	61.19%	54.20%	1.29	1.29
2021	22.09%	21.47%	41.67%	33.47%	1.53	1.56
2022	37.59%	36.72%	-40.40%	-47.22%	1.30	1.32

Table 5.5: Summary Results for Inversly Volatility Weighted Portfolio

Dataset	Annualized Risk		Annualized Return		Diversification Index	
	K-01	K-02	K-01	K-02	K-01	K-02
2019-2022	27.84%	26.51%	22.41%	18.25%	1.36	1.38
2019	17.91%	17.97%	42.69%	35.37%	1.55	1.60
2020	37.10%	36.09%	48.36%	41.61%	1.28	1.28
2021	19.78%	17.54%	40.34%	28.08%	1.57	1.67
2022	33.05%	32.07%	-30.05%	-40.26%	1.32	1.35

for 2022. In 2022, most stock returns were negative, and both traditional and data-driven portfolio construction approaches did not yield positive returns against risk for that year. But, SRO provides almost 4% positive returns for the K-01 portfolio (consisting of the 12 assets with the highest mean return) with approximately 22% risk.

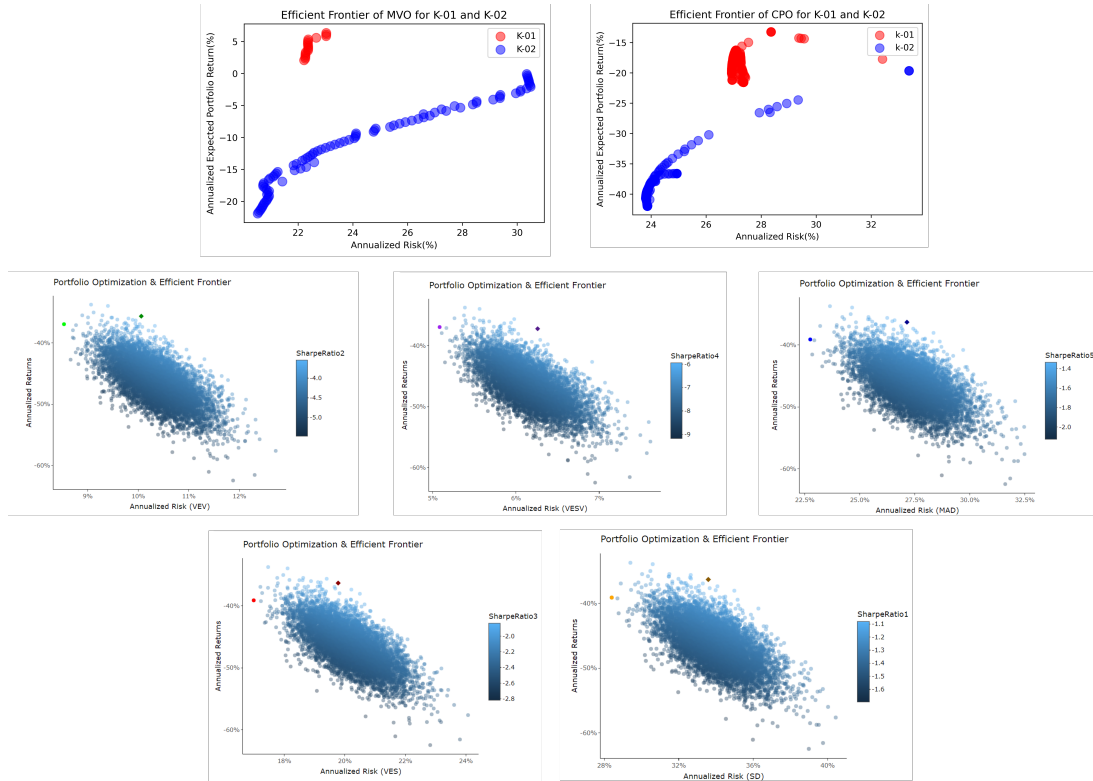


Figure 5.11: Efficient Frontiers using mean-variance, constraint portfolio optimization and data-driven approach (2022)

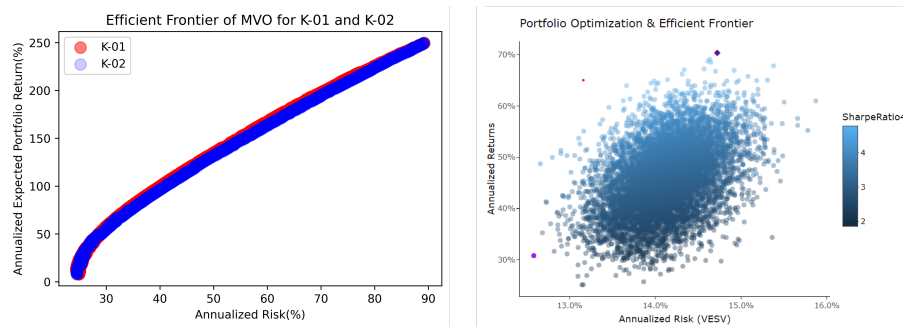


Figure 5.12: Efficient Frontiers for technological stock-based portfolios using mean-variance optimization (left) and data-driven approach (right) for 2020

For other years and the combined data of all four years, traditional models such as MVO and CPO perform better in the efficient frontier plots, compared to data-driven models. In 2020, MVO performed with high returns at a given annualized risk. These results are shown in the efficient frontiers ¹ of the Figures 5.11 and 5.12.

5.3.2 Cryptocurrency-based PO

Traditional portfolio optimization models like MVO performed better than data-driven strategies for a highly volatile cryptocurrency portfolio in a 10-month period in 2022, which can be observed from Figure 5.13.

CPO did not show an efficient frontier for both cryptocurrency portfolios on the same scale, but gave a positive indication of the relationship between annualized risk and returns when observed separately. Additionally, the S&P CME Bitcoin Future-based approach performed better than the Crypto20-based approach, as shown in the Figure 5.14.

5.3.3 Key Findings

The traditional models for portfolio construction, such as MVO, SRO, and CPO solve objective functions for portfolio weight optimization. In contrast, recent studies focus solely on risk measures to determine the minimum risk weights from random weight samples. They apply this approach to construct resilient portfolios for regular stocks, cryptocurrencies, and

¹Efficient frontiers represent the set of optimal portfolios that offer the highest expected return for a given level of risk, or the lowest risk for a given level of expected return. These portfolios are plotted on a graph where the x-axis represents risk (typically measured by standard deviation or variance) and the y-axis represents expected return.

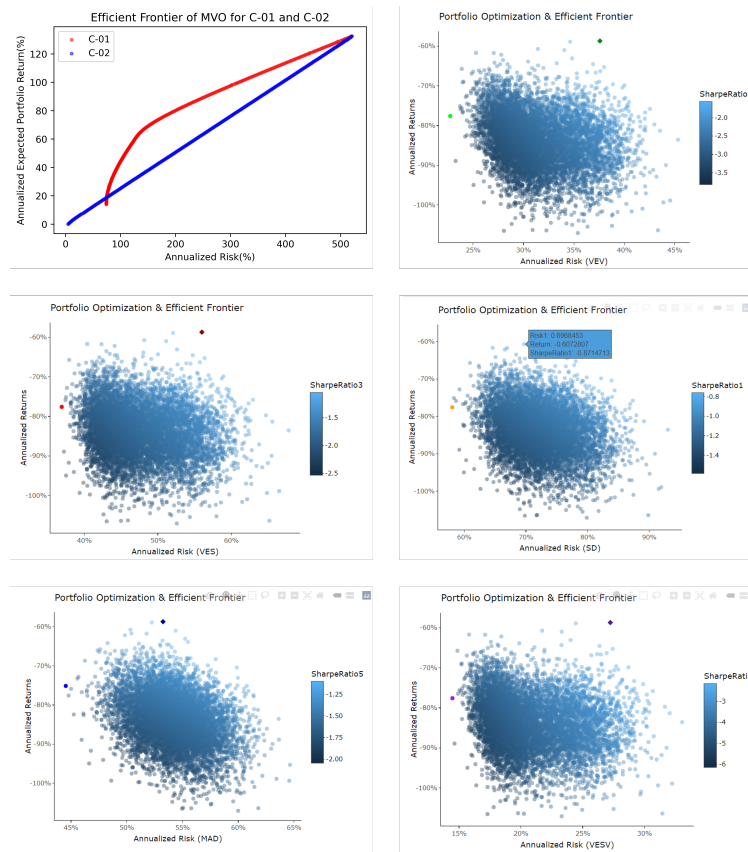


Figure 5.13: Efficient Frontiers for cryptocurrency-based portfolios using mean-variance optimization and data-driven approach

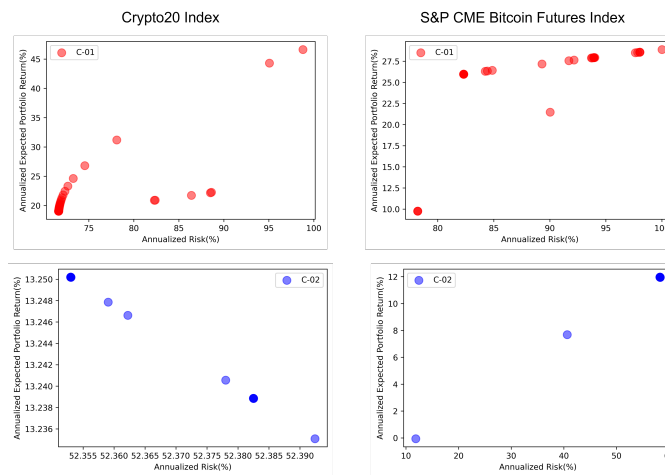


Figure 5.14: Efficient Frontiers with crypto index using constrained portfolio optimization

their combination using hourly data. Although a maximum number of methods provide positive returns against annualized risk, they fail to identify portfolio weights for positive returns for short-term data with negative mean returns. In such cases, SRO can identify a portfolio with positive portfolio returns by selecting assets with high mean returns based on clustering. For

certain years like 2020, approaches such as MVO perform relatively well. From the highly volatile daily cryptocurrency data considered in this study, it seems that traditional portfolio optimization provides positive returns. The study uses the Crypto20 and S&P CME bitcoin futures crypto indexes for performance analysis, and the results show that the latter provides better insight into the market than the former index when measuring beta. Moreover, when constructing portfolios using constrained portfolio optimization, the S&P CME bitcoin futures-based approach yields better results.

5.4 Investigation of Clustering Algorithms for Diversification

This study builds on previous research by examining the effectiveness of various clustering techniques in introducing diversification within different PO methods (see Chapter 4.4). It also explores lesser-known algorithms, such as DBSCAN and AP, in the financial context. The results from the experiments are outlined in the discussion below.

5.4.1 Clustering on EW and IVW

Although diversification with clustering algorithms improved the performance of simple EW and IVW, they still failed in risky years like 2022. The AP clustering algorithm produced the best annual profits in 2020, but all clustering algorithms failed to generate positive returns in 2022.

Table 5.6: Performance using different clustering algorithms for E.W and I.V.W constructions

Year	Clustering Methods	Equal Weighted				Inverse Volatility Weighted			
		K-01		K-02		K-01		K-02	
		Annual Risk	Annual Return	Annual Risk	Annual Return	Annual Risk	Annual Return	Annual Risk	Annual Return
2019	kmeans	15.91%	25.90%	15.49%	23.81%	14.19%	27.14%	13.37%	23.11%
2020		38.74%	50.37%	37.75%	33.89%	36.59%	44.04%	34.84%	26.76%
2021		17.27%	48.76%	16.24%	42.48%	15.07%	43.54%	13.50%	37.12%
2022		28.87%	-18.75%	29.60%	-23.33%	25.02%	-6.90%	26.59%	-14.82%
2019	Hierarchical	15.91%	25.90%	15.36%	23.31%	14.19%	27.14%	12.96%	22.65%
2020		38.95%	63.19%	37.78%	43.59%	36.52%	55.96%	34.47%	34.80%
2021		16.64%	48.50%	15.76%	42.01%	13.90%	42.02%	13.14%	36.09%
2022		28.55%	-18.02%	29.73%	-24.10%	23.29%	-2.47%	25.67%	-12.72%
2019	DBSCAN	16.56%	27.80%	16.73%	25.70%	15.25%	29.08%	15.29%	26.17%
2020		38.56%	39.53%	38.81%	37.13%	34.75%	35.52%	34.69%	32.62%
2021		16.49%	42.30%	15.95%	40.02%	15.14%	38.76%	14.26%	36.41%
2022		27.37%	-16.38%	21.95%	-69.18%	24.11%	-6.13%	15.02%	-22.80%
2019	AP	17.68%	24.74%	16.99%	20.88%	15.35%	25.83%	14.68%	21.59%
2020		42.83%	85.54%	42.25%	68.44%	39.55%	77.08%	40.09%	56.36%
2021		20.54%	59.04%	20.65%	51.37%	18.46%	56.34%	16.96%	44.75%
2022		28.55%	-18.02%	29.73%	-24.10%	23.29%	-2.47%	25.67%	-12.72%

Table 5.6 presents the performance of different clustering algorithms in conjunction with EW and IVW portfolio construction techniques in different years. EW and IVW are not reliable in risky years like 2022.

This study delves into the application of MVO and SRO in portfolio management. Specifically, the influence of clustering techniques on these strategies is investigated, analyzing their efficacy in enhancing diversification and identifying profitable opportunities.

5.4.2 Clustering with MVO and CPO

Figures 5.15 and 5.16 showcase MVO's effectiveness with diverse clustering algorithms for 2020 and 2022, revealing consistent portfolio outcomes on the efficient frontier. For 2022, the clustering algorithms demonstrate comparable performance characteristics.

Diverse clustering techniques notably boosted conventional MVO performance, resulting in significant 2020 profitability. Impressively, this diversification sustained profits even during the high-risk year of 2022. For 2022, Figure 5.16 highlights efficient frontiers using k-means clustering.

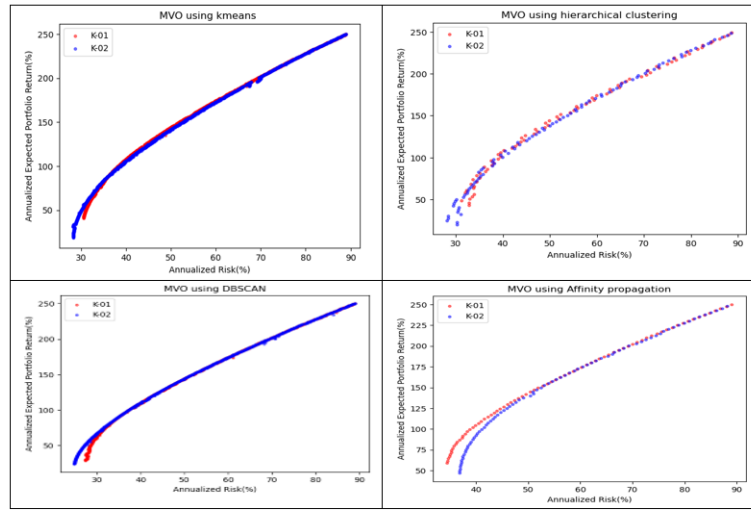


Figure 5.15: MVO with different clustering algorithms for 2020

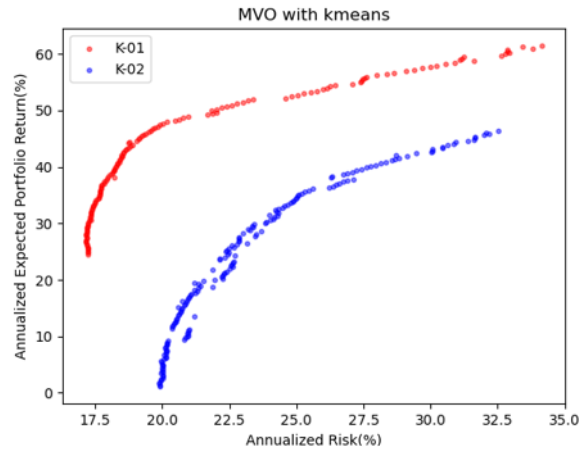


Figure 5.16: MVO with kmeans clustering algorithm for 2022

A. Constrained Portfolio Optimization (CPO)

CPO extends MVO by accommodating risk-averse investors' constraints, shaping a customized framework. Clustering algorithms showed similar performance in MVO, but CPO indicated less favorability for AP, especially during 2020 and 2022. Observations from Figure 5.17 highlight suboptimal performance of CPO with AP in 2020 compared to k-means, suggesting limited effectiveness of AP for CPO-based portfolios.

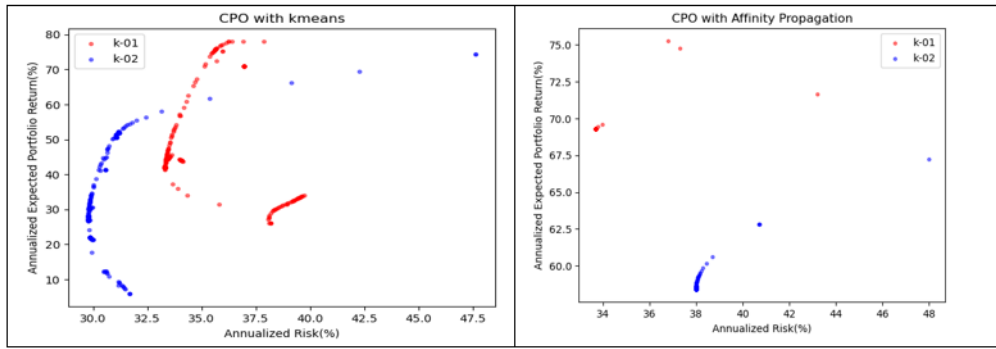


Figure 5.17: CPO with kmeans clustering algorithm(left) and AP(right) in 2020

The study revealed that in the tumultuous market of 2022, portfolios created using DBSCAN and k-means clustering algorithms outperformed those using hierarchical and AP algorithms (Figure 5.18).

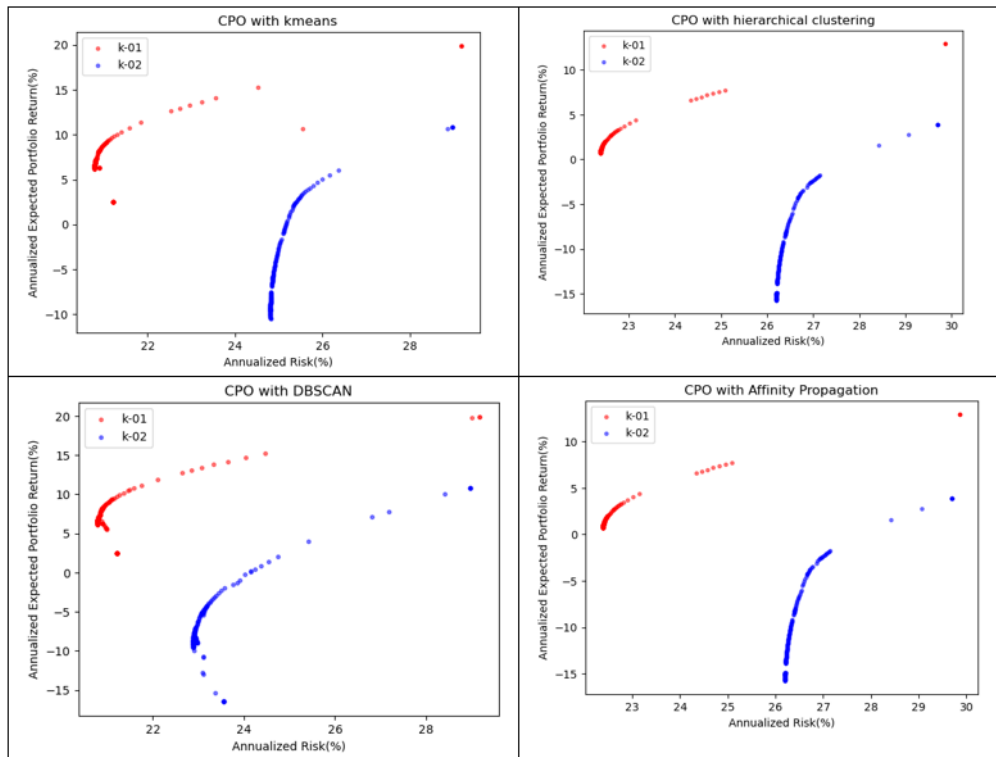


Figure 5.18: CPO with clustering algorithms for 2022

5.4.3 Clustering with SRO

SRO maximizes risk-adjusted return using excess return to volatility ratio as a key metric. The application of diverse clustering techniques significantly improved the performance of SRO. Notably, for 2020, k-means demonstrated superior risk-adjusted profitability compared to other clustering algorithms, as observed in Table 5.7.

Table 5.7: SRO with different clustering techniques

Year	Clustering Methods	SHARPE RATIO OPTIMIZATION					
		K-01			K-02		
		Annual Risk	Annual Return	Sharpe Ratio	Annual Risk	Annual Return	Sharpe Ratio
2019	kmeans	13.77%	42.63%	2.88	14.41%	42.51%	2.74
2020		58.88%	171.31%	2.86	14.41%	42.51%	2.74%
2021		15.16%	57.62%	3.62	13.36%	51.93%	3.66
2022		19.29%	46.57%	2.26	26.58%	38.65%	1.34
2019	Hierarchical	13.77%	42.63%	2.88	13.26%	40.06%	2.80%
2020		58.81%	171.08%	2.83	59.99%	172.80%	2.83
2021		14.42%	55.06%	3.61	14.25%	54.97%	3.65
2022		19.45%	46.92%	2.26	25.54%	36.87%	1.33
2019	DBSCAN	14.68%	45.02%	2.86	14.91%	43.96%	2.75
2020		57.40%	166.76%	2.85	55.57%	161.31%	2.85
2021		16.07%	60.67%	3.59	14.21%	54.04%	3.65
2022		19.05%	45.99%	2.26	25.01%	36.35%	1.33
2019	AP	14.65%	44.08%	2.8	14.81%	36.87%	2.29
2020		62.33%	180.88%	2.85	59.61%	172.66%	2.85
2021		16.89%	63.45%	3.58	17.65%	55.12%	2.95
2022		19.45%	46.93%	2.26	25.54%	36.87%	1.33

5.4.4 Clustering with Data-Driven(DD) Approach

In traditional portfolio strategies, non-normality in financial data is overlooked. Data-driven approaches consider this, optimizing portfolios with data-centric risk metrics. This study investigates enhancing these techniques with diversification and recommends effective clustering methods for robust portfolio optimization, particularly in volatile markets.

A. Data-Driven PO Without Clustering

For 2022, the data-driven approach without clustering algorithms showed limited profitability, as illustrated in Figure 5.19 but integrating clustering algorithms not only bolstered performance that year but also improved overall performance across other years. This study high-

lights varying effectiveness among clustering algorithms in enhancing portfolio construction techniques, particularly in risk-prone years like 2022, emphasizing a data-driven assessment.

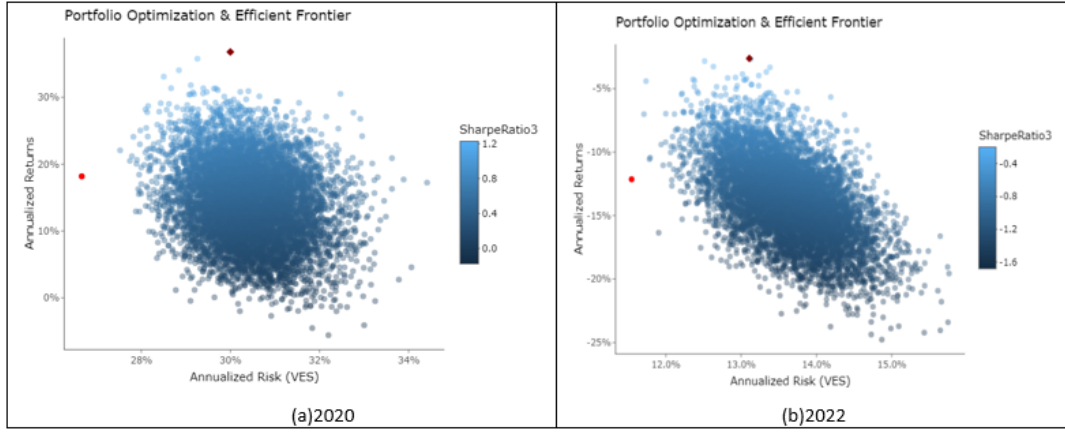


Figure 5.19: Data-driven portfolio optimization without clustering

B. Data-Driven PO With Clustering

The integration of diversification through various clustering techniques significantly enhanced the profitability of the data-driven portfolio construction technique.

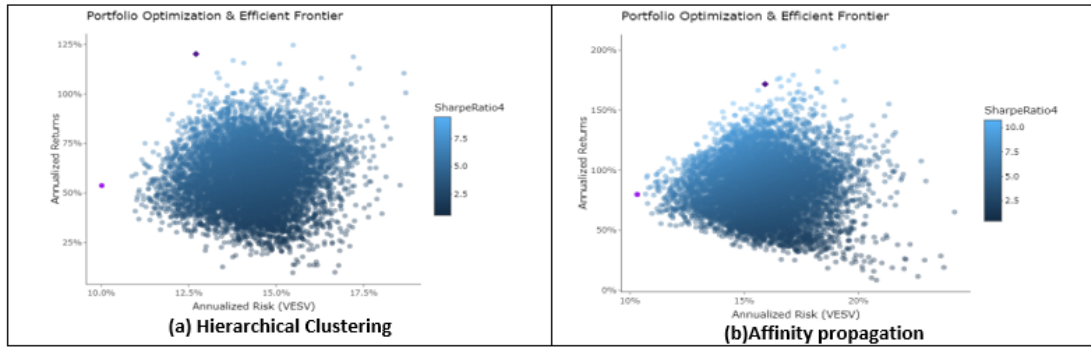


Figure 5.20: Data-driven approach with Hierarchical (left) and AP(right) clustering algorithms for 2020

For 2020, the utilization of AP and hierarchical clustering showcased enhanced performance in diversification compared to alternative algorithms, as supported by Figure 5.20, with AP particularly excelling in constructing profitable portfolios during that year.

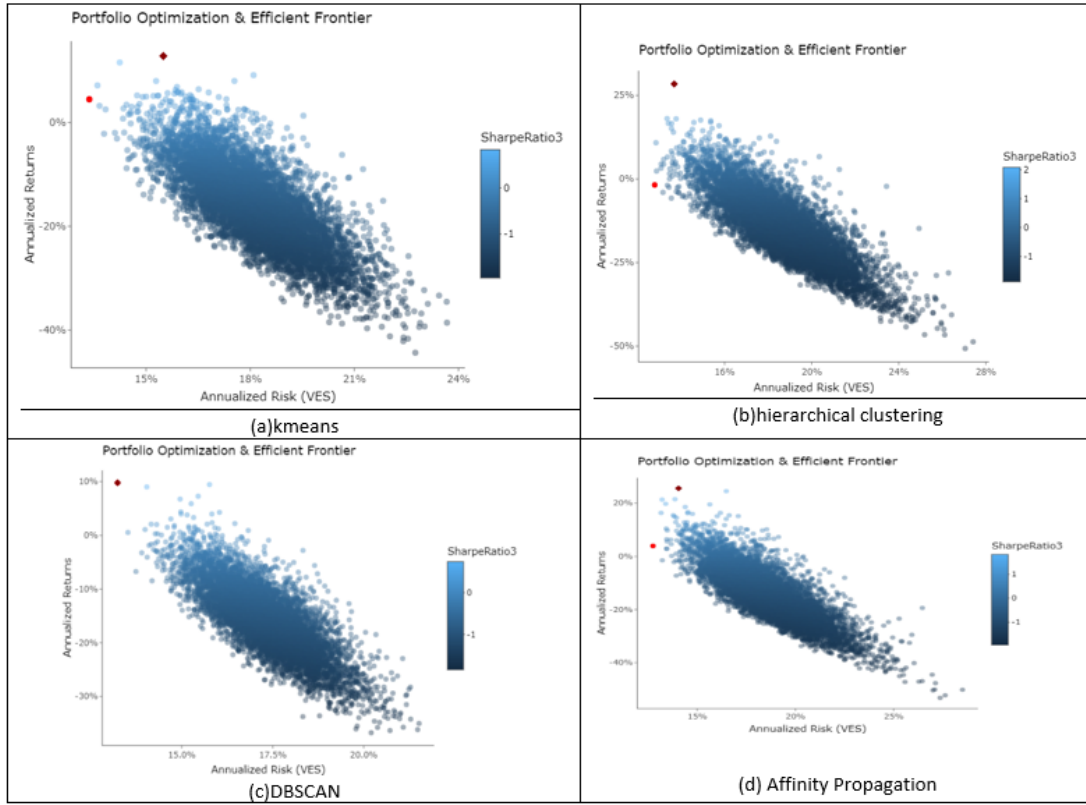


Figure 5.21: Data-driven approach with different clustering algorithms for 2022

Figure 5.21 illustrates the superior performance of hierarchical clustering and AP-based portfolio construction for 2022 within a data-driven approach, highlighting the potential of AP for enhanced diversification and portfolio robustness.

5.4.5 Key Findings

This study underscores the pivotal role of diversification through clustering techniques in improving portfolio resilience. Analyzing various clustering methods and portfolio construction approaches reveals that portfolio performance is contingent on the choice of clustering technique and prevailing market conditions. Optimal diversification, crucial for portfolio resilience, necessitates a careful selection of assets from disparate clusters. Different clustering methods exhibit distinct strengths, with AP clustering synergizing effectively with traditional approaches like EW and IVW optimization during profitable periods. MVO demonstrates consistent effectiveness across clustering techniques, while k-means clustering stands out in SRO during

favorable market conditions. The integration of data-driven portfolio construction techniques, notably in conjunction with AP clustering, remarkably enhances portfolio performance. These insights hold pragmatic implications for both investors and portfolio managers, emphasizing the advantages of diversification facilitated by clustering techniques in constructing resilient portfolios.

	Traditional PO methods	Data-driven PO approaches
Statistical Features	Traditional methods often overlook critical factors such as non-normality, skewness, and kurtosis in financial data distributions.	Data-driven approaches account for significant statistical features, including non-normality, skewness, and kurtosis, within financial data distributions.
Robustness	Traditional methodologies demonstrate less robustness compared to approaches based on data-driven risk measures.	Data-driven risk measures offer enhanced profitability relative to the traditional approaches grounded in standard deviation.
Effectiveness through clustering	1. The inclusion of clustering algorithms improves the efficacy of traditional portfolio optimization techniques. 2. K-means and other algorithms are effective for MVO, with K-means and DBSCAN being applied to CPO, and K-means and AP for SRO contexts.	1. Clustering techniques substantially improve performance, particularly during challenging periods. 2. Hierarchical clustering and AP-based strategies enhance resilience within data-driven portfolio optimization approaches.

Table 5.8: Comparison between Traditional and Data-driven PO Methods

Table 5.8 presents a comparison between traditional and data-driven PO approaches in a tabular format.

Chapter 6

Conclusion, Future Works, and Contributions

This research has examined advanced methodologies for predictive accuracy and resilient portfolio construction in financial markets through a series of studies. Each study has addressed distinct but complementary aspects of recent innovations in traditional models, ML/DL, and data science techniques applied to price prediction and PO, along with the novel proposal of new hybridized architectures for these two application areas.

6.1 Conclusions

The first study in this thesis introduced innovative encoder-decoder-based RNN architectures designed for predicting financial prices across various types of assets under different market conditions. The proposed AE-GRU architecture, which combines an encoder-decoder framework with GRU, outperformed AE-LSTM across various market scenarios. Experiments using diverse stocks, the S&P index, and bitcoin showed that AE-GRU consistently surpasses AE-LSTM, even after extensive hyperparameter tuning of AE-LSTM. Training on data from significant market events, AE-GRU demonstrated superior predictive accuracy for financial assets.

The second study of this thesis work focused on constructing resilient portfolios using deep

learning and data-driven optimization techniques. By accurately forecasting stock prices with BiLSTM and BiGRU, the research showed that BiGRU performed better than BiLSTM in challenging market conditions. Using 2023 data, both predicted and actual, this study demonstrated that diversified portfolios significantly outperform non-diversified ones. Specifically, VES-based measures excelled in non-diversified portfolios, while VEV-based measures were found to be superior in diversified portfolios. These findings highlighted that diversification, especially with AP, and selecting appropriate risk measures enhanced overall portfolio performance.

Traditional portfolio construction models like MVO, SRO, and CPO optimize portfolio weights by solving objective functions, while recent studies focus on risk measures to determine minimum risk weights from random samples. Although many methods delivered positive returns against annualized risk, they struggled with short-term data with negative mean returns. This third study showed that the S&P CME bitcoin futures crypto index offers better market insights than the Crypto20 index, particularly in constructing portfolios using constrained optimization. Data-driven PO techniques were found as more robust compared to the traditional techniques but the profitability increased significantly for traditional approaches when the stock selection was diversified with a clustering technique. This research introduced a new line of inquiry into identifying specific clustering techniques to enhance diversification within traditional PO approaches.

The previous study highlighted Diversification as a critical factor in enhancing portfolio resilience. Through various clustering techniques, such as AP and k-means, the fourth study of this thesis revealed that optimal diversification involves carefully selecting assets from disparate clusters. This study highlighted the critical role of diversification through specific clustering techniques in enhancing portfolio resilience. The performance of portfolios depends on the choice of clustering method and market conditions. MVO consistently performed well across clustering methods, while k-means and DBSCAN clustering excelled in SRO under favorable conditions. Integrating data-driven portfolio construction techniques with AP clustering significantly improved portfolio performance, providing valuable insights for investors and portfolio managers on the benefits of clustering-based diversification, especially in crisis periods.

Most of the research reported in this thesis led to five peer-reviewed publications, including three conference papers and two journal articles. The research mainly focused on developing effective portfolio optimization techniques using machine learning for various financial assets with fluctuating volatilities and diverse statistical features. This thesis designed ML/DL/DS models resulting in hyperparameter-tuned novel architectures for financial price prediction and resilient portfolio optimization techniques.

6.2 Future Works

The experiments done for this thesis relied on limited features of the financial assets. Factors related to PO, such as interest rates, transaction costs, fees, commissions, *etc.* were not considered. If these factors were included, the research could be applied in practical scenarios. Additionally, detailed hyperparameter tuning has been challenging due to limited computational resources. To address this issue, GPU computing will be utilized in future experiments.

Further research will integrate advanced statistical approaches into ML algorithms for advancing price prediction and PO. Moreover, ongoing investigations explore the combination of DL-based approaches with evolutionary computation algorithms for PO, which are planned to be disseminated in a conference/journal.

ML algorithms in this research worked with the empirical covariance matrix for use in clustering algorithms. There are multiple correlation approaches that can be tested to extend the existing research in confirming the interaction of different types of correlations with the clustering algorithms for introducing profitable diversification. Though the LSTM and GRU algorithms used in this research are introduced with autoencoder-based novel modification, how the models are trained leaves room for further extension of the research. The data used for enhancing predictive accuracy contained the adjusted closing prices, but the interaction of stocks with the highly correlated basket of other stocks was not considered in this price prediction. Often the price prediction of a particular stock relies on the fluctuation of values of other related stocks. In this case, the reliability should be measured and consideration of such

reliability in price prediction opens a new avenue for research. An ongoing research project considers the incorporation of data-driven correlation approaches to measure the reliability and changes in that reliability among different financial assets in price prediction.

DL approaches when used with a large amount of data, create problems in execution with limited computational resources. For doing detailed research with DL models, powerful GPUs need to be used for the execution of models with multiple powerful layers. In this thesis, two encoder-decoder-based RNN architectures were proposed, which showed high predictive accuracy. There is a new interest in TN-based architectures which needs further exploration for price prediction in conjunction with the encoder-decoder-based architectures. A new direction of research will involve GPU computing to deal with such issues. Hybridization of ML/DL algorithms with these powerful TN and informer networks (IN) algorithms will open new avenues of research in the near future.

The research in PO is being heavily explored with ML/DL-based algorithms. However, the novel approaches with TN/IN architectures require heavy usage of computational resources. Moreover, the incorporation of factors such as transaction cost, commissions, fees, *etc.* in PO research will increase the practicality of the research outcome in real-world scenarios. For a new research project involving PO, these factors will be carefully integrated.

6.3 List of Published Contributions

My research works and contributions during the MSc program resulted in the following publications:

1. **Joy Dip Das**, Sulalitha Bowala, Ruppa K Thulasiram, Aerambamoorthy Thavaneswaran, *"Resilient Portfolio Optimization using Traditional and Data-Driven Models for Cryptocurrencies and Stocks"*: 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), 26-30 June 2023, Torino, Italy, doi: 10.1109/COMPSAC57700.2023.00204 (went through double-blind peer review)
2. **Joy Dip Das**, Sulalitha Bowala, Ruppa K Thulasiram, Aerambamoorthy Thavaneswaran,

- "Portfolio Diversification with Clustering Techniques"*: Proceedings of the 2023 IEEE Symposium Series on Computational Intelligence (SSCI), 05-08 December 2023, Mexico City, Mexico, doi: 10.1109/SSCI52147.2023.10371938 (went through double-blind peer review)
3. Sulalitha Bowala, Aerambamoorthy Thavaneswaran, Ruppa Thulasiram, Thimani Ranathunga, **Joy Dip Das**, *"High Frequency Data-Driven Dynamic Portfolio Optimization for Cryptocurrencies"*: Proceedings of the 2023 IEEE Symposium Series on Computational Intelligence (SSCI), 05-08 December 2023, Mexico City, Mexico, doi: 10.1109/SSCI52147.2023.10371938 (went through double-blind peer review)
 4. **Joy Dip Das**, Sulalitha Bowala, Ruppa K Thulasiram, Aerambamoorthy Thavaneswaran, *"Hybrid Data-Driven and Deep Learning Based Portfolio Optimization"*: Journal of Mathematical Finance, Vol.14 No.3, August 2024, 271-310, doi: 10.4236/jmf.2024.143016
 5. **Joy Dip Das**, Ruppa K Thulasiram, Christopher Henry, Aerambamoorthy Thavaneswaran, *"Encoder-Decoder Based LSTM and GRU Architectures for Stocks and Cryptocurrency Prediction"*: Journal of Risk and Financial Management, Vol.17, issue no.5, 2 May 2024, (Special Issue Machine Learning Applications in Finance) doi: <https://doi.org/10.3390/jrfm17050200>

Appendix A

Algorithms

A.1 LSTM

LSTM is an RNN architecture that can handle long-term dependencies in the data. [28] address the issue of vanishing gradient faced by the basic RNN. From Figure A.1, we can see that it comprises specialized memory units that can maintain and update information over extended sequences¹. These memory units, called cells, contain three gating mechanisms: input, output, and forget gates. The input gate oversees the selection of data to be stored in the cell, the forget gate governs the determination of information to be discarded, and the output gate manages the information to be emitted from the cell.

The LSTM equations:

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

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(C_t)$$

f_t is the forget gate, governing the information to be omitted from the cell state. i_t denotes

¹<https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af>

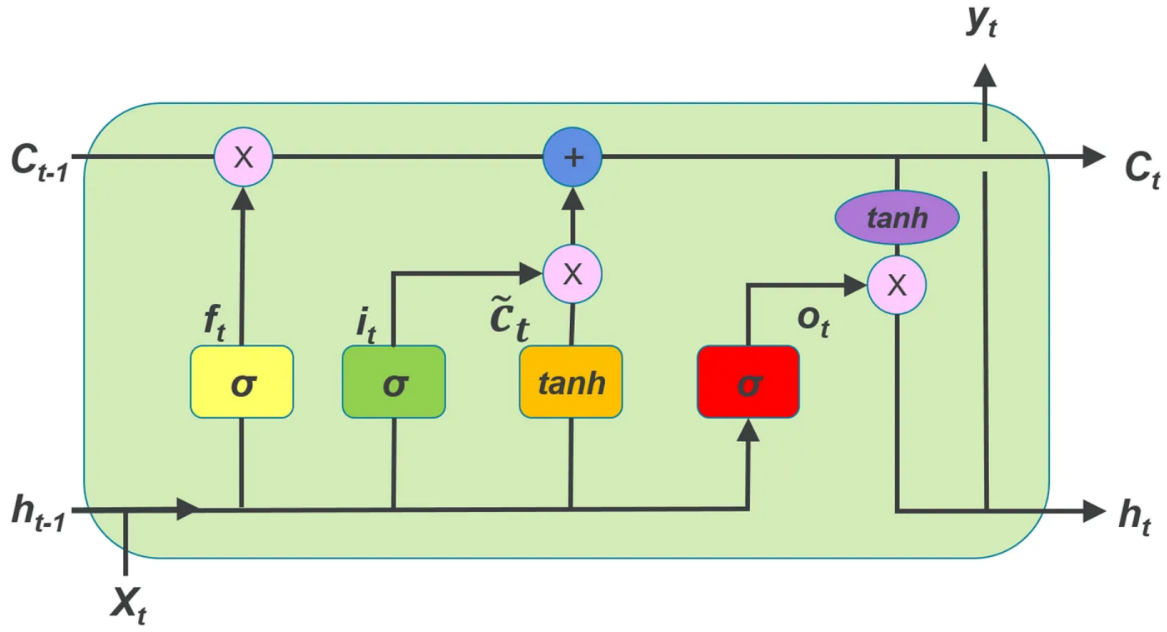


Figure A.1: LSTM architecture

the input gate, deciding which values to revise in the cell state. $\tilde{C}_t$ signifies the candidate cell state, presenting new candidate values for the cell state. C_t represents the cell state, storing long-term information. o_t indicates the output gate, managing the information for output from the cell state. h_t corresponds to the hidden state, encompassing the short-term information for output.

A.2 GRU

The GRU, a prevalent RNN algorithm in stock price prediction, exhibits a unique architecture with gating mechanisms regulating information flow, enhancing long-term dependency capture in input data². Core components, including reset and update gates, enable effective context utilization as seen in Figure A.2. GRU's computational efficiency, attributed to its simpler two-gate structure, contributes to faster training times compared to LSTM (see [29]).

²https://www.researchgate.net/publication/355705028/figure/fig2/AS:108387081234023101635426347531/GRU-structure-diagram_W640.jpg

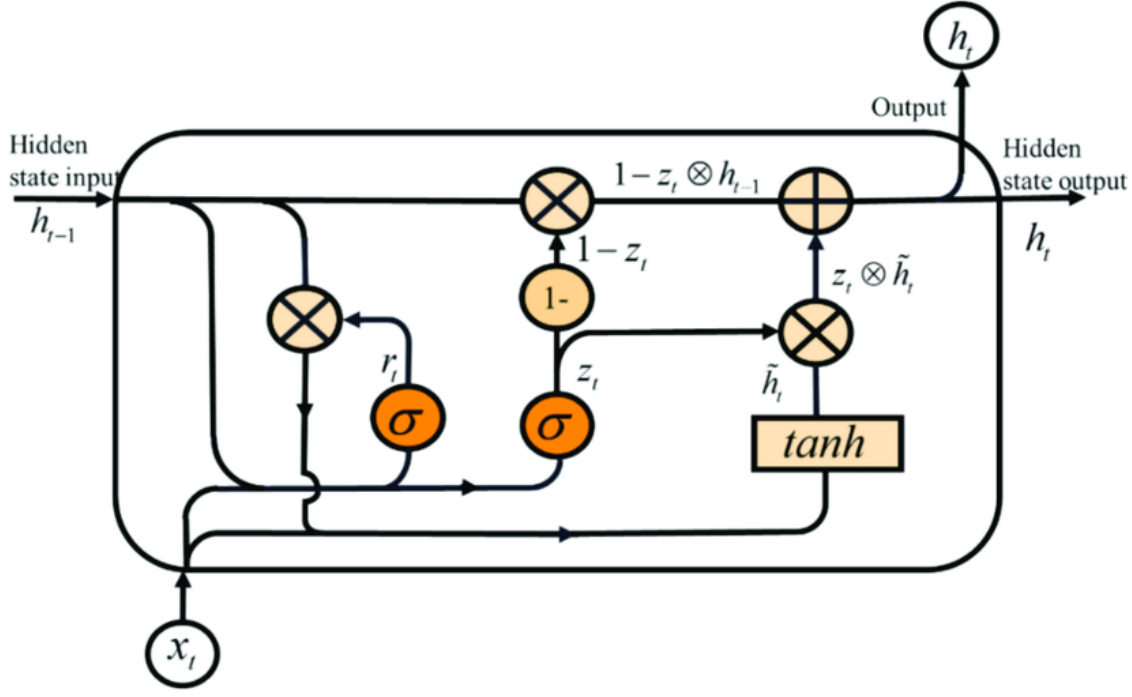


Figure A.2: GRU architecture

The following equations can explain GRU:

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

$$\tilde{h}_t = \tanh(r_t \odot U h_{t-1} + W x_t)$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}$$

In the context of the GRU model, r_t serves as the reset gate, while z_t functions as the update gate. The vector $\tilde{h}_t$ represents a candidate activation vector. The functions σ and $\tanh$ denote the sigmoid and hyperbolic tangent functions, respectively. The weight matrices W_r , U_r , W_z , U_z , W , and U play pivotal roles in the model.

LSTMs employ a more intricate structure with separate gates for input, forget, and output, enabling them to preserve long-term dependencies in the input sequence by controlling the flow of information. In contrast, GRUs feature a simplified architecture by merging the functionality of the input and forget gates into a single "update gate," resulting in fewer parameters and computational resources. While LSTMs maintain separate cell states and hidden states for memory management, GRUs combine them into a single vector, which may slightly reduce

memory capacity. Both LSTM and GRU units have demonstrated effectiveness in various applications, and the choice between them often depends on factors such as model complexity and computational resources.

Bibliography

- [1] D. Appel and M. Grabinski. The origin of financial crisis: A wrong definition of value. *Portuguese Journal of Quantitative Methods*, 2:33–51, 2011.
- [2] The Bank for International Settlements. Basel III monitoring report. <https://www.bis.org/bcbs/publ/d531.pdf>, 2021. Accessed on 04.06.2023.
- [3] Y. Choueifaty and Y. Coignard. Toward maximum diversification. *The Journal of Portfolio Management*, 35(1):40–51, 2008.
- [4] M. Metghalchi, J. Marcucci, and Y. H. Chang. Are moving average trading rules profitable? evidence from the european stock markets. *Applied Economics*, 44(12):1539–1559, 2012.
- [5] D. Banerjee. Forecasting of indian stock market using time-series arima model. In *2014 2nd international conference on business and information management (ICBIM)*, pages 131–135. IEEE, 2014.
- [6] D. G. McMillan. Non-linear predictability of uk stock market returns. *Oxford Bulletin of Economics and Statistics*, 65(5):557–573, 2003.
- [7] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7:77–91, 1952.
- [8] F. J. Fabozzi, F. Gupta, and H. M. Markowitz. The legacy of modern portfolio theory. *The Journal of Investing*, 11(3):7–22, 2002.
- [9] J. Mossin. Equilibrium in a capital asset market. *Econometrica*, 34(4):768–783, 1966.
- [10] J. Lintner. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *Review of Economics and Statistics*, 47(1):13–37, 1965.

- [11] W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, 19(3):425–442, 1964.
- [12] E. F. Fama and K. R. French. The capital asset pricing model: Theory and evidence. *Journal of Economic Perspectives*, 18(3):25–46, 2004.
- [13] B. Solnik. International arbitrage pricing theory. *The Journal of Finance*, 38(2):449–457, 1983.
- [14] W. Cheung. The black–litterman model explained. *Journal of Asset Management*, 11:229–243, 2010.
- [15] G. V. Pai. *Metaheuristics for Portfolio Optimization: An Introduction Using MATLAB*. John Wiley & Sons, 2017.
- [16] W. F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.
- [17] M. E. A. G. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman. Nse stock market prediction using deep-learning models. *Procedia computer science*, 132:1351–1362, 2018.
- [18] U. M. Sirisha, M. C. Belavagi, and G. Attigeri. Profit prediction using arima, sarima, and lstm models in time series forecasting: A comparison. *IEEE Access*, 10:124715–124727, 2022.
- [19] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications*, 197:116659, 2022.
- [20] Y. E. Cakra and B. D. Trisedya. Stock price prediction using linear regression based on sentiment analysis. In *2015 International Conference on Advanced Computer Science and Information Systems (ICACISIS)*, pages 147–154, 2015.

- [21] Z. K. Lawal, H. Yassin, and R. Y. Zakari. Stock market prediction using supervised machine learning techniques: An overview. In *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pages 1–6. IEEE, 2020.
- [22] K. M. Hindrayani, T. M. Fahrudin, R. P. Aji, and E. M. Safitri. Indonesian stock price prediction including covid19 era using decision tree regression. In *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 344–347. IEEE, 2020.
- [23] J. Chen. Analysis of bitcoin price prediction using machine learning. *Journal of Risk and Financial Management*, 16(1):51, 2023.
- [24] A. M. Khedr, I. Arif, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intelligent Systems in Accounting, Finance and Management*, 28(1):3–34, 2021.
- [25] T. B. Shahi, A. Shrestha, A. Neupane, and W. Guo. Stock price forecasting with deep learning: A comparative study. *Mathematics*, 8(9):1441, 2020.
- [26] Y. Ma, R. Han, and W. Wang. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165:113973, 2021.
- [27] E. Pintelas, I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas. Investigating the problem of cryptocurrency price prediction: a deep learning approach. In *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16*, pages 99–110. Springer, 2020.
- [28] A. Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.
- [29] S. Yang, X. Yu, and Y. Zhou. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, pages 98–101. IEEE, 2020.

- [30] G. Y. Ban, N. El Karoui, and A. E. Lim. Machine learning and portfolio optimization. *Management Science*, 64(3):1136–1154, 2018.
- [31] W. Chen, H. Zhang, M. K. Mehlawat, and L. Jia. Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing*, 100:1–18, 2021.
- [32] F. D. Paiva, R. T. N. Cardoso, G. P. Hanaoka, and W. M. Duarte. Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, 115:635–655, 2019.
- [33] Z. Zhang, S. Zohren, and S. Roberts. Deep learning for portfolio optimization. *The Journal of Financial Data Science*, 2(4):8–20, 2020.
- [34] C. Y. Liou, W. C. Cheng, J. W. Liou, and D. R. Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.
- [35] S. Gu, B. Kelly, and D. Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450, 2021.
- [36] K. Xia, J. Huang, and H. Wang. Lstm-cnn architecture for human activity recognition. *IEEE Access*, 8:56855–56866, 2020.
- [37] A. A. Pierre, S. A. Akim, A. K. Semenyó, and B. Babiga. Peak electrical energy consumption prediction by arima, lstm, gru, arima-lstm and arima-gru approaches. *Energies*, 16(12):4739, 2023.
- [38] M. L. G. Rubell, N. Selby, and A. Taparia. An efficient stock price prediction mechanism using multivariate sequential lstm autoencoder, 2023. Preprint, 21 February 2023. Version 1. Available online: <https://www.researchsquare.com/article/rs-2599921/v1> (accessed on 2 March 2024).
- [39] H. K. Cao and B. T. Nguyen. Delafo: An efficient portfolio optimization using deep neural networks. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-*

Asia Conference, PAKDD, Singapore, May 11–14, Proceedings, Part I 24, pages 623–635.

Springer International Publishing, 2020.

- [40] F. Rundo, F. Trenta, A. L. di Stallo, and S. Battiato. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.
- [41] X. Zhong and D. Enke. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67:126–139, 2017.
- [42] P.P.S. Kohli, S. Zargar, S. Arora, and P. Gupta. Stock prediction using machine learning algorithms. In *Applications of Artificial Intelligence Techniques in Engineering*, pages 1–38. Springer, Singapore, 2019.
- [43] O. A. Awoye. *Markowitz Minimum Variance Portfolio Optimization Using New Machine Learning Methods*. Doctoral dissertation, University College London (UCL), 2016.
- [44] J. Du. Mean-variance portfolio optimization with deep learning-based forecasts for integrated stocks. *Expert Systems with Applications*, 201:117005, 2022.
- [45] V. Tola, F. Lillo, M. Gallegati, and R. N. Mantegna. Cluster analysis for portfolio optimization. *Journal of Economic Dynamics and Control*, 32(1):235–258, 2008.
- [46] J. Behera, A. K. Pasayat, H. Behera, and P. Kumar. Prediction based mean-value-at-risk portfolio optimization using machine learning regression algorithms for multi-national stock markets. *Engineering Applications of Artificial Intelligence*, 120:105843, 2023.
- [47] F. A. Ibanez. Diversified spectral portfolios: An unsupervised learning approach to diversification. *The Journal of Financial Data Science*, 2023. Advance online publication.
- [48] X. Li. A regularized k-means autoregressive shrinkage method to portfolio optimization. *Highlights in Business, Economics and Management*, 5:192–204, 2023.
- [49] B. Aslam, R. A. Bhuiyan, and C. Zhang. Portfolio construction with k-means clustering algorithm based on three factors. In *MATEC Web Conf.*, volume 377, page 02006, 2023. Curtin Global Campus Higher Degree by Research Colloquium (CGCHDRC 2022).

- [50] K. S. Pranata, A. A. S. Gunawan, and F. L. Gaol. Development clustering system idx company with k-means algorithm and dbscan based on fundamental indicator and esg. *Procedia Computer Science*, 216:319–327, 2023.
- [51] Z. Fang and C. Chiao. Research on prediction and recommendation of financial stocks based on k-means clustering algorithm optimization. *Journal of Computational Methods in Sciences and Engineering*, 21(5):1081–1089, 2021.
- [52] J. Doering, R. Kizys, A. A. Juan, A. Fito, and O. Polat. Metaheuristics for rich portfolio optimization and risk management: Current state and future trends. *Operations Research Perspectives*, 6:100121, 2019.
- [53] Y. Wang and T. Aste. Dynamic portfolio optimization with inverse covariance clustering. *Expert Systems with Applications*, 213:118739, 2023.
- [54] Z. Hu, Y. Zhao, and M. Khushi. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1):9, 2021.
- [55] W. Jiang. Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, 184:115537, 2021.
- [56] M. Nikou, G. Mansourfar, and J. Bagherzadeh. Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management*, 26:164–174, 2019.
- [57] M. Nabipour, P. Nayyeri, H. S. Jabani, and A. Mosavi. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, 8:150199–150212, 2020.
- [58] J. Jang and N. Seong. Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory. *Expert Systems with Applications*, page 119556, 2023.

- [59] A. Fernández and S. Gómez. Portfolio selection using neural networks. *Computers & Operations Research*, 34(4):1177–1191, 2007.
- [60] A. S. Uysal, X. Li, and J. M. Mulvey. End-to-end risk budgeting portfolio optimization with neural networks. *Annals of Operations Research*, pages 1–30, 2023.
- [61] L. Yu, S. Wang, and K. K. Lai. Neural network-based mean–variance–skewness model for portfolio selection. *Computers & Operations Research*, 35(1):34–46, 2008.
- [62] S. Yang. Deep reinforcement learning for portfolio management. *Knowledge-Based Systems*, 278:110905, 2023.
- [63] V. M. Ngo, H. H. Nguyen, and P. Van Nguyen. Does reinforcement learning outperform deep learning and traditional portfolio optimization models in frontier and developed financial markets? *Research in International Business and Finance*, 65:101936, 2023.
- [64] Y. Ma, W. Wang, and Q. Ma. A novel prediction based portfolio optimization model using deep learning. *Computers & Industrial Engineering*, page 109023, 2023.
- [65] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble. Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market. In *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pages 228–234, 2019.
- [66] Q. Jiang, C. Tang, C. Chen, X. Wang, and Q. Huang. Stock price forecast based on LSTM neural network. In J. Xu, F. Cooke, M. Gen, and S. Ahmed, editors, *Proceedings of the Twelfth International Conference on Management Science and Engineering Management. ICMSEM 2018*, pages 403–412. Springer, 2019.
- [67] D. Sheth and M. Shah. Predicting stock market using machine learning: Best and accurate way to know future stock prices. *International Journal of Systems Assurance Engineering and Management*, 14:1–18, 2023.
- [68] C. Qi, J. Ren, and J. Su. Gru neural network based on ceemdan–wavelet for stock price prediction. *Applied Sciences*, 13(12):7104, 2023.

- [69] P. Gupta, Y. Gao, R. Wang, and E. Zhou. Stock prediction based on optimized lstm and gru models. *Scientific Programming*, page 4055281, 2021.
- [70] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza. Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm: a deep learning approach. *Fractal and Fractional*, 7(2):203, 2023.
- [71] J. Kim, S. Kim, H. Wimmer, and H. Liu. A cryptocurrency prediction model using lstm and gru algorithms. In *2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD)*, pages 37–44. IEEE, 2021.
- [72] T. A. Phan-Thi, N. T. Dinh, S. H. Thanh, and B. T. Nguyen. Portfolio optimization for long-term investment: An empirical study. In *New Trends in Intelligent Software Methodologies, Tools and Techniques*, pages 230–241. IOS Press, 2023.
- [73] M. Ali, D. M. Khan, H. M. Alshanbari, and A. A. A. H. El-Bagoury. Prediction of complex stock market data using an improved hybrid emd-lstm model. *Applied Sciences*, 13(3):1429, 2023.
- [74] Z. Fang, X. Ma, H. Pan, G. Yang, and G. R. Arce. Movement forecasting of financial time series based on adaptive lstm-bn network. *Expert Systems with Applications*, 213:119207, 2023.
- [75] H. Zhao, X. Li, J. Xu, X. Fu, and J. Chen. Financial time series data prediction by combination model adaboost-knn-lstm. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Gold Coast, Australia, 2023. IEEE.
- [76] N. Kwak and D. Lim. Financial time series forecasting using adaboost-gru ensemble model. *Journal of the Korean Data And Information Science Society*, 32:267–281, 2021.
- [77] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu. Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science*, 131:895–903, 2018.

- [78] H. Song and H. Choi. Forecasting stock market indices using recurrent neural network based hybrid models: Cnn-lstm, gru-cnn, and ensemble models. *Applied Sciences*, 13(7):4644, 2023.
- [79] A. John and T. Latha. Stock market prediction based on deep hybrid rnn model and sentiment analysis. *Automatika*, 64(4):981–995, 2023.
- [80] A. Petrovic, I. Strumberger, T. Bezdan, H. S. Jassim, and S. S. Nassor. Cryptocurrency price prediction by using hybrid machine learning and beetle antennae search approach. In *2021 29th Telecommunications Forum (TELFOR)*, pages 1–4. IEEE, 2021.
- [81] M. Sharma and H. S. Shekhawat. Portfolio optimization and return prediction by integrating modified deep belief network and recurrent neural network. *Knowledge-Based Systems*, 250:109024, 2022.
- [82] P. Singh, M. Jha, M. Sharaf, M. A. Elmeligy, and T. R. Gadekallu. Harnessing a hybrid cnn-lstm model for portfolio performance: A case study on stock selection and optimization. *IEEE Access*, 11:104000–104015, 2023.
- [83] T. Cui, N. Du, X. Yang, and S. Ding. Multi-period portfolio optimization using a deep reinforcement learning hyper-heuristic approach. *Technological Forecasting and Social Change*, 198:122944, 2024.
- [84] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, ..., and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [85] S. Bowala and J. Singh. Optimizing portfolio risk of cryptocurrencies using data-driven risk measures. *Journal of Risk and Financial Management*, 15(10):427, 2022.
- [86] A. Thavaneswaran, Y. Liang, N. Yu, A. Paseka, and R. K. Thulasiram. Novel data-driven resilient portfolio risk measures using sign and volatility correlations. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1742–1747. IEEE, July 2021.

- [87] D. Forsyth. *Applied machine learning*. Springer, 2019.
- [88] P-N Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.
- [89] D. Dueck. *Affinity propagation: clustering data by passing messages*. University of Toronto Toronto, ON, Canada, 2009.
- [90] C. C. Chang, Z. T. Lin, W. W. Koc, C. Chou, and S. H. Huang. Affinity propagation clustering for intelligent portfolio diversification and investment risk reduction. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 145–150, 2016.
- [91] D. Kitanovski, M. Mirchev, I. Chorbev, and I. Mishkovski. Cryptocurrency portfolio diversification using network community detection. In *2022 30th Telecommunications Forum (TELFOR)*, pages 1–4, 2022.
- [92] M. Pirani, P. Thakkar, P. Jivrani, M. H. Bohara, and D. Garg. A comparative analysis of arima, gru, lstm and bilstm on financial time series forecasting. In *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, pages 1–6. IEEE, 2022.
- [93] S. Salimath, T. Chatterjee, T. Mathai, P. Kamble, and M. Kolhekar. Prediction of stock price for indian stock market: A comparative study using lstm and gru. In *Advances in Computing and Data Sciences: 5th International Conference, ICACDS 2021, Nashik, India, April 23–24, 2021, Revised Selected Papers, Part II 5*, pages 292–302. Springer, 2021.
- [94] N. Cakici, C. Fieberg, D. Metko, and A. Zaremba. Do anomalies really predict market returns? new data and new evidence. *Review of Finance*, 28(1):1–44, 2024.
- [95] P. T. Yamak, L. Yujian, and P. K. Gadosey. A comparison between arima, lstm, and gru for time series forecasting. In *Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence*, pages 49–55, dec 2019.