



Predicting Stock Performance from Financial Ratios Using Machine Learning

Prepared By

Name: Marwa O. Essa

ID No: 243002281

Submitted in fulfillment of the requirements for the degree Master of
Science in Artificial Intelligence

At

College of Informatics

Midocean University

UAE

Supervisor Name: Dr. Ashraf Elsayed

College of Informatics, Midocean University, Comoros

Email:.....

Acknowledgements

I extend my deepest gratitude to my supervisor, Dr. Ashraf Elsayed, for his invaluable assistance, unwavering encouragement, and insightful critiques throughout my research journey. His steadfast commitment to academic excellence and meticulous attention to detail have significantly shaped the quality and depth of this dissertation.

I am also sincerely grateful to the Teaching Staff Members at the College of Informatics, Midocean University, whose guidance, resources, and encouragement have been invaluable to my academic development.

To my family, I express my deepest love and appreciation. Your unconditional love, patience, and sacrifices have been the foundation of my resilience and success. Without your constant encouragement, this achievement would not have been possible.

Abstract

Using machine learning for stock prediction has gained a lot of interest recently. Many studies show that machine learning algorithms can forecast stock movements effectively by using historical data. However, most current methods mainly focus on short-term predictions based on past stock prices and technical indicators, such as the RSI and MACD. In this research, we take a different approach. We use 41 quarters of financial statement data to explore the possibility of predicting long-term stock returns based on fundamental indicators. The model includes important financial ratios, such as profitability, leverage, efficiency, liquidity, and market value ratios, to create a predictive framework. We trained three machine learning algorithms: RF, LSTM, and GRU. We selected portfolios by selecting stocks with higher predicted returns using these machine learning models. Portfolios were constructed by selecting stocks with higher predicted returns according to these models. The constructed portfolios outperformed the TASI benchmark in the Saudi equity market, achieving higher returns and portfolio values. The findings emphasize the advantages of employing financial ratios in predicting the top 5 high-returning stocks in a mid-to-long-term investment situation, achieving over 33% higher excess returns for all models with lower risk profiles. Also, the selection of the top 3 high-return stocks achieved over 78% relative returns with a reasonable risk level, proving the effectiveness of applying ML techniques based on financial ratios for stock prediction.

List of Publications

Table of Contents

Acknowledgements	iii
Abstract.....	v
List of Publications	v
Table of Contents	vi
List of Abbreviations	viii
List of Figures.....	ix
List of Tables	ix
Chapter 1 Introduction	1
1.1 Problem Definition.....	1
1.2. Thesis Objectives	1
1.3 Thesis Contribution.....	1
1.4 Thesis Organization	2
Chapter 2 Overview.....	3
2.1 Introduction	3
2.2 Essential Concepts	4
2.2.1 Random Forest (RF).....	4
2.2.2 Long Short-Term Memory (LSTM)	5
2.2.3 Gate Recurrent Unit (GRU)	7
Chapter 3 Related Work.....	11
3.1 Introduction.....	11
3.2 History.....	11
Chapter 4 Proposed Approach	13
4.1 METHODOLOGY.....	13
4.1.1 Data Description.....	13
4.1.2 Financial Ratios.....	14

4.1.3 Data Preprocessing.....	15
4.1.4 Calculation of the Target Variable – Relative Return	16
4.1.2. Data Split and Moving Time Period	17
4.1.4 Evaluation Metrics	18
4.1.3. Training	20
Chapter 5 Experimental Results and Discussion.....	23
Chapter 6 Conclusion and Future Work.....	25
References	26

List of Abbreviations

- **Acc:** Accuracy
- **GRU:** Gated Recurrent Unit
- **RF:** Random Forest
- **LSTM:** Long Short-Term Memory
- **MACD:** Moving Average Convergence Divergence
- **ML:** Machine Learning
- **OHLCV:** Open, High, Low, Close, Volume
- **RNNs:** Recurrent Neural Networks
- **ROA:** Return on Assets
- **ROC:** Return on Capital
- **ROCE:** Return on Common Equity
- **ROE:** Return on Equity
- **RF:** Random Forest
- **RSI:** Relative Strength Index
- **MAE:** Mean Absolute Error
- **SGD:** Stochastic Gradient Descent
- **TASI:** Tadawul All Share Index

List of Figures

2.1	Random Forest	21
2.2	The architecture of LSTM	23
2.3	The architecture of GRU	23
4.1	Rolling Window Data Splitting for Random Forest Training	23
4.2	Rolling Window Data Splitting for LSTM and GRU Training	23
4.4	Importance of features of RF model.....	23
5.1	Predicted vs Actual Price.....	23

List of Tables

4.1	13 Financial Ratios	21
4.2	Hyperparameters of LSTM model	23
4.3	Hyperparameters of GRU model	23
5.1	Models Evaluation	23

Chapter 1 Introduction

1.1 Problem Definition

Despite the increasing use of artificial intelligence in financial markets, most stock prediction models focus mainly on technical analysis. They often overlook the valuable insights found in corporate financial statements. Traditional machine learning models also have difficulty capturing complex, non-linear relationships between fundamental financial indicators and stock price behavior. This research aims to fill this gap by creating a machine learning-based model that predicts stock trends using fundamental financial data. Specifically, the research looks at how key profitability ratios, such as ROA, ROC, ROE, and ROCE, affect stock price movements. The main goal is to improve the accuracy, clarity, and reliability of long-term stock price predictions by using AI-driven methods.

1.2. Thesis Objectives

This research proposes a new method for estimating the true value of stock returns. It uses information from financial statements and applies machine learning techniques. The goal is to improve the decision-making abilities of investors and financial analysts. This approach offers a data-driven way to evaluate stocks for the long term.

1.3 Thesis Contribution

Evaluating machine learning methods for long-term stock prediction based on fundamental analysis. We do so by comparing the prediction performance of three advanced machine learning methods based on fundamental analysis using financial statements.

1.4 Thesis Organization

This thesis is structured around six chapters that cover major concerns of the research. Chapter 1: Introduction offers a background to the research issue, objectives, and contributions. It states the background of the problem, formulates the problem, defines the objectives, and presents the significance of stock performance prediction with machine learning and financial ratios. Chapter 2: Overview covers the background and basic concepts concerning research. It describes principles of data mining, specifically their application in financial forecasting, and provides a description of the theoretical foundations that underline the methodological strategy of this thesis. Chapter 3: Related Work surveys the literature on stock prediction techniques, including traditional statistical models and contemporary machine learning strategies. It focuses on previous research on technical indicators and financial ratios, indicates gaps in literature, and positions the current research within the context of general research. Chapter 4: Proposed Approach describes the research methodology adopted in this research, encompassing data collection, preprocessing steps, moving time-period strategy, and the applied machine learning models RF, LSTM, and GRU. Chapter 5: Experimental Results and Discussion reports the experimental configuration, evaluation metrics, and results of the proposed models. The results are benchmarked against the TASI and assessed based on excess return, Top-K stock selection, and portfolio score, followed by a discussion of their implications for investment strategy in practical terms. Finally, Chapter 6: Conclusion and Future Work summarizes the major findings and contributions of this research, highlighting the success of integrating financial ratios with machine learning for long-term stock forecasting. Moreover, it specifies the limitations of the research and suggests directions for further research to build on and improve the current research.

Chapter 2 Overview

2.1 Introduction

The main reason for predicting stock price movements is the chance for profit. Since financial markets began, investors have looked for ways to forecast stock performance to find profitable opportunities. This involves buying stocks expected to increase in value and selling them for a profit. Over the years, many stock prediction methods have appeared, but their consistency and reliability are still debated.

Stock prediction methods can generally be grouped into three main categories:

Fundamental Analysis. This approach predicts stock prices by looking at the financial health of companies using publicly available data, including financial statements and economic indicators. It aims to estimate a stock's true value.

Technical Analysis. This technique relies only on past market data such as prices and trading volumes. In contrast to fundamental analysis, technical analysis assumes that all public information is already included in a stock's price. It is based on three main ideas:

- Market action discounts everything.
- Prices are moving in trends.
- History repeats itself.

Sentiment Analysis. This method assesses investor sentiment by examining news articles, analyst reports, and online commentary related to specific stocks or the overall market. It seeks to identify the current mood or emotional tone influencing investor behavior.

Among these methods, technical analysis and sentiment analysis are mainly used for short-term predictions, for days or even hours, while fundamental analysis is usually applied for mid- to long-term forecasts, spanning quarters or years. Recently, ML and data mining techniques into financial modeling have gained significant attention. Many existing ML-based stock prediction studies focus on short-term forecasting using technical indicators or sentiment data. However, these models often neglect real-world factors like transaction costs, which can affect the practical use and reliability of their results.

This research aims to fill this gap by examining machine learning methods for long-term stock prediction based on fundamental analysis. The research

specifically investigates the predictive power of financial statement data in estimating future stock performance. To achieve this, we compare the prediction performance of three machine learning models trained on fundamental financial features. Our dataset includes quarterly financial data from 100 companies listed on the TASI (Tadawul All Share Index) between 2015 and 2025. We assess model performance by ranking 70 stocks based on their predicted relative returns and building portfolios accordingly. The actual returns of these portfolios serve as the main measure for evaluating prediction accuracy and model effectiveness.

2.2 Essential Concepts

In [2] [6], Random Forest (RF) was shown to have better portfolio performance. However, LSTM and GRU models excel in modeling temporal and sequential dependencies. Due to these strengths, we chose these models in this work.

2.2.1 Random Forest (RF)

Random Forest (RF) is a powerful ensemble learning technique that has become popular for stock selection and financial forecasting. Initially introduced by Breiman in 2001 [12] and further developed in conjunction with Cutler [10], RF is based on the concept of decision trees. While training, the model develops several decision trees, each of which is grown using a randomly drawn subset of data and input variables. Such randomness helps to promote tree diversity and avoids the possibility of overfitting, thus leading to improved generalization capability of the model.

By using bootstrap aggregation (bagging), RF also stabilizes predictions and reduces variance, rendering it a reliable and efficient means of stock performance prediction. In stock selection, RF uses historical financial and market data to recognize patterns that could predict stocks' future trend or value. These predictive patterns, as shown in Figure 2.1, offer useful recommendations to investors and portfolio managers in making sound decisions.

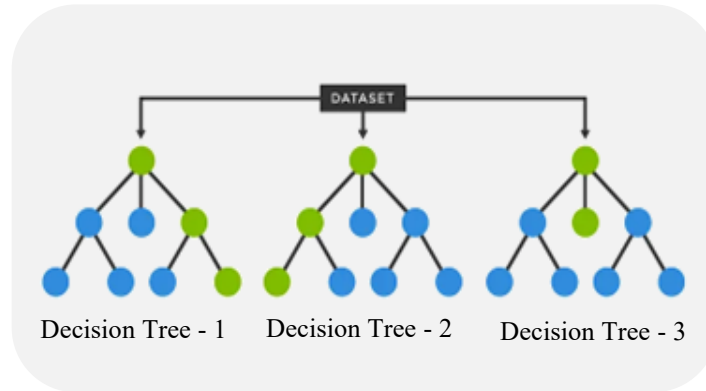


Figure 2.1 Random Forest [3]

2.2.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber in 1997 [13] as a solution to the vanishing gradient problem in RNNs. The vanishing gradient effect happens when the gradient of the loss function vanishes quickly when performing backpropagation through time, making it difficult for standard RNNs to learn and capture long-term dependencies.

LSTM expands the RNN architecture by adding a memory cell that can retain information for long periods [4], thus keeping valuable contextual information intact even when there are large intervals between related inputs. It makes LSTM very suitable for modeling sequential data. Unlike standard RNNs, LSTM adds a gating mechanism controlling information flow into and out of the memory cell. Three gates specifically include input gate, forget gate, and output gate control how information is added, forgotten, and propagated. Each of the gates is controlled by a sigmoid activation function, which produces numbers between 0 and 1, effectively modulating the amount of information flow. As a result of these characteristics, LSTMs have been universally used in many fields such as natural language processing, speech recognition, and financial time series prediction, where long-term temporal dependencies need to be captured. It is defined by the following equations:

The input gate activation

$$i_t = \sigma(w_i \cdot [h_{t-1}x_t] + b_i) \quad (2.1)$$

Where:

x_t is the input at time t

h_{t-1} is the hidden state at time t-1

w_i is the weight matrices

b_i is the bias vector

The forget gate controls the flow of information from the memory cell into the hidden state. It determines what information to keep or discard from the memory cell. It is defined by the following equations:

$$f_t = \sigma(w_f \cdot [h_{t-1}x_t] + b_f) \quad (2.2)$$

Where:

f_t is the forget gate activation

w_f is the weight matrices

b_f is the bias vector.

The output gate controls the flow of information from the memory cell to the hidden state and output. It is defined by the following equations:

$$o_t = \sigma(w_o \cdot [h_{t-1}x_t] + b_o) \quad (2.3)$$

Where:

O_t is the output gate activation

w_o the weight matrices

b_o is the bias vector

The memory cell has the following equations:

$$\tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

Where:

$\tilde{C}_t$ is the memory cell

w_c is the weight matrices

b_c is the bias vector.

The output of the LSTM at time t is given by:

$$h_t = 0t * \tanh(c_t) \quad (2.5)$$

Where:

h_t is the hidden state at time t .

While training, the LSTM model reduces the difference between the forecasted output and the genuine target values by optimizing a loss function, generally using gradient descent–based optimizers like SGD or Adam.

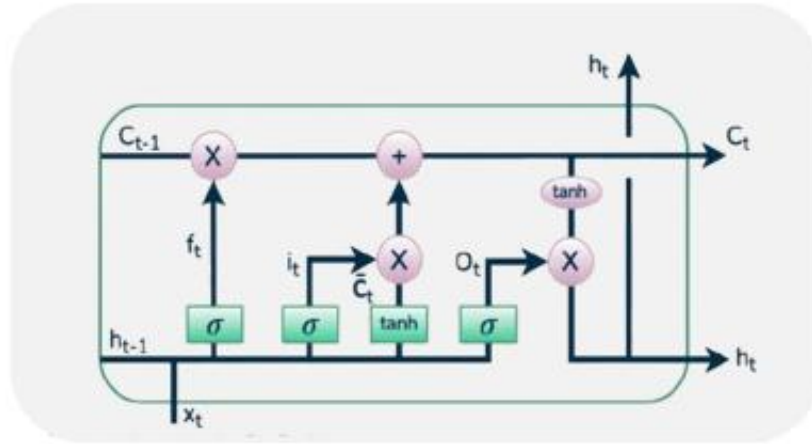


Figure 2.2 The architecture of LSTM [14]

2.2.3 Gate Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture proposed by Kyunghyun Cho et al. [9] in 2014. It is very similar to the Long Short-Term Memory (LSTM) network [15], frequently achieving the same performance. Nevertheless, the GRU is more computationally efficient,

needing fewer parameters, which decreases training complexity as well as memory requirements.

GRU is particularly suited for handling sequential data by facilitating the easy flow of information through time steps. This makes it very effective in scrutinizing historical stock data for dominant patterns and trends that can potentially affect future performance. Therefore, GRU serves a central role in stock forecasting models, facilitating more accurate predictions and enabling better-informed investment choices. The Gated Recurrent Unit (GRU) is also ubiquitous in sequence modeling and financial time series forecasting, where long-term temporal dependencies must be modeled. It is characterized by the following equations:

$$r_t = \sigma(w_r \cdot [h_{t-1}x_t] + b_r) \quad (2.6)$$

Where:

r_t is the reset gate activation

w_r are the weight matrices

b_r is the bias vector

The **update gate** determines how much of the past information should be carried forward to the current state:

$$z_t = \sigma(w_z \cdot [h_{t-1}x_t] + b_z) \quad (2.7)$$

Where:

z_t is the update gate activation

w_z are the weight matrices

b_z is the bias vector.

The candidate hidden state computes the new content to be added, regulated by the reset gate:

$$\tilde{h}_t = \tanh(w_h \cdot [r_t * h_{t-1}, x_t] + b_h) \quad (2.8)$$

Where:

$\tilde{h}_t$ is the candidate hidden state

w_h are the weight matrices

b_h is the bias vector.

The hidden state of the GRU at time t is given by blending the previous hidden state with the candidate hidden state, controlled by the update gate:

$$h_t = (1 - z_t) * n_{t-1} + z_t * h_t \quad (2.9)$$

Where : h_t is the updated hidden state at time t .

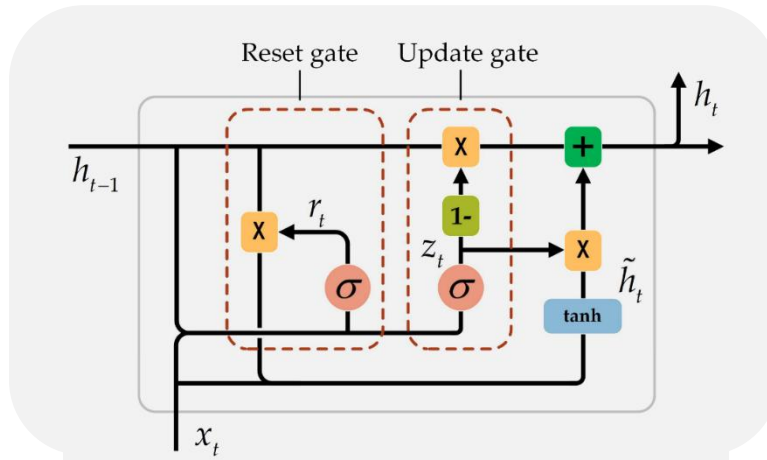


Figure 2.3 The architecture of GRU [15]

The Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two widely used variants of recurrent neural networks developed to overcome the gradient-related difficulties in traditional RNNs and effectively capture long-term dependencies in sequential data. LSTMs employ a more complex architecture with three gates: input, forget, and output; and maintain a separate memory cell that enables the model to store and regulate information across long temporal intervals. In contrast, GRUs adopt a simpler design with only two gates, the reset and update gates, and do not maintain a separate memory cell; instead, they combine the hidden state and memory into a single representation. This simplification makes

GRUs computationally more efficient, requiring fewer parameters and shorter training times, while still achieving performance comparable to LSTMs in many applications. In general, LSTMs are better suited for modeling very long-term dependencies, whereas GRUs tend to deliver similar results on shorter sequences or smaller datasets.

Chapter 3 Related Work

3.1 Introduction

Stock price forecasting has always been a finance and data science problem since markets are dynamic, volatile, and multi-factorial. Previously, stock prediction was largely reliant on statistical and econometric models that involved histories of past prices. However, with swift advancements in ML and DL in recent years, efforts have been made to utilize more sophisticated methods involving both technical indicators and fundamentals.

3.2 History

In recent years, numerous studies have explored the application of machine learning techniques in stock prediction. Gupta, H., & Jaiswal, A. [1] in 2024 introduced research on stock forecasting using deep learning and statistical models. The survey motive is to check various deep learning and statistical model techniques for stock price forecasting that are Moving Averages, ARIMA which are statistical techniques and LSTM, RNN, CNN, and FULL CNN which are deep learning models. The result of all the models is different and the best result comes from the LSTM model and we have also seen how Full CNN outweighs the pre-process CNN. This model helps the investors to know when to sell and buy the stock prices. As is also known how the stock prices are affected by the climatic or geographical changes, company profit or loss, change in geographical seasons, loss in the region, inflation, competition in the market etc. Additionally, Tsai, P.-F., Gao, C.-H., & Yuan, S.-M. [2] in 2023 delved into deep learning models for this purpose. They introduced a paper about, Stock selection using machine learning based on financial ratios, found that successful utilization of RF, FNN, GRU, and FinGAT models holds significant managerial implications for enhancing portfolio management strategies. On the other hand, Kunathananon, T. [3] in 2022 introduced Estimating stock price based on information from financial statement using machine learning approaches. This research introduces a tool for stock market investors and institutions constructed from Long Short-Term Memory (LSTM) for predicting stock prices. By effectively analyzing financial statements and stock market data, the results indicate that LSTM can effectively capture complex relationships in the data and

predict stock prices. This research highlights the potential of LSTM as a valuable and innovative tool for investors and institutions in the stock market. In 2022, Z.Y. Lu and S.M. Yuan [6] employed an FNN model with 18 financial ratios and sector dummy variables to predict potential stock returns for the next quarter. This approach improved prediction accuracy by incorporating sectors as dummy variables. In a historical context, the works of Heo, J., & Yang, J. Y. [7] in 2016 this research about stock price prediction based on financial statements using SVM. Han, S., & Chen, R.-C. [8] in 2007. The research used SVM with financial statement analysis for prediction of stocks.

Chapter 4 Proposed Approach

4.1 METHODOLOGY

Three machine learning models: Random Forest (RF), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), are used in this research to forecast stock performance using ratios from financial statements. The models leverage past quarterly data to predict future stock returns. The models are evaluated using the Mean Absolute Error (MAE) and other metrics such as Excess Return, Top-k Precision, and Portfolio Score.

4.1.1 Data Description

Within the Saudi stock market (TASI), our stock pool comprises 51 companies selected based on sectoral representation. These include the Energy sector, which consistently holds the highest market capitalization; the Materials sector, recognized as the largest non-oil contributor; and the Capital Goods sector. The TASI index serves as the benchmark for evaluating portfolio performance, providing a standard against which to compare the returns of model-generated portfolios.

The selected stocks span the period from Q1 2015 to Q1 2025, covering 41 quarters of financial data. Based on this comprehensive dataset, we developed four predictive models to estimate the relative return of stocks. The goal is to identify the top 5 and top 10 performing stocks each quarter to construct optimized investment portfolios. Portfolio returns are then compared to the TASI index to evaluate relative performance.

Additionally, OHLCV (Open, High, Low, Close, Volume) data was collected for the same stock pool to support realistic investment simulations. Opening prices were used to represent the feasible purchase price of each stock at the beginning of the investment period. This approach ensures that return calculations reflect actual market conditions, offering a more accurate assessment of portfolio performance.

The dataset was compiled through web scraping from the Saudi Exchange (Tadawul) [15] and other reputable financial data providers [16][17].

4.1.2 Financial Ratios

We use 13 financial ratios as input features for the predictive models, consistent with those employed in research [5]. These ratios, along with their respective calculation formulas, are presented in Table 1.

Ratio	Calculation	Description
Current Ratio	$\text{Current Assets} \div \text{Current Liabilities}$	Measures a company's ability to pay short-term obligations with short-term assets.
Debt to Equity	$\text{Total Debt} \div \text{Shareholders' Equity}$	Shows the proportion of debt used relative to equity financing.
Long Term Debt/Capital	$\text{Long-Term Debt} \div (\text{Long-Term Debt} + \text{Shareholders' Equity})$	Indicates the proportion of long-term debt in the total capital structure.
Asset Turnover	$\text{Revenue} \div \text{Average Total Assets}$	Measures how efficiently assets are used to generate revenue.
Inventory Turnover	$\text{Sales} \div \text{Inventories}$	Shows how often inventory is sold and replaced over a period.
Receivable Turnover	$\text{Net Credit Sales} \div \text{Receivables}$	Indicates how effectively receivables are collected.
Days Sales in Receivables	$(\text{Receivables} \div \text{Net Credit Sales}) \times 365$	Measures the average collection period for receivables.
Book Value Per Share	$(\text{Shareholders' Equity} - \text{Preferred Equity}) \div \text{Total Outstanding Shares}$	Represents the equity value allocated to each outstanding share.
Gross Profit Margin	$(\text{Revenue} - \text{COGS}) \div \text{Revenue}$	Shows the percentage of revenue remaining after direct costs.
EBITA Margin	$\text{EBITA} \div \text{Revenue}$	Measures profitability before interest, taxes, and amortization.
EBIT Margin	$\text{EBIT} \div \text{Revenue}$	Measures profitability before interest and taxes.
Net Income Margin	$\text{Net Income} \div \text{Revenue}$	Shows the percentage of revenue remaining as net profit.
Return on Assets	$\text{Net Income} \div \text{Total Assets}$	Measures efficiency in using assets to generate profit.
Return on Capital	$\text{EBIT} \div (\text{Total Debt} + \text{Shareholders' Equity})$	Evaluates returns generated from total capital employed.
Return on Equity	$\text{Net Income} \div \text{Shareholders' Equity}$	Measures return generated on total shareholders' equity.
Return on Common Equity	$\text{Net Income Available to Common Shareholders} \div \text{Common Equity}$	Focuses on returns to common shareholders, excluding preferred equity.

Table 4.1 13 Financial Ratios

4.1.3 Data Preprocessing

The original data set had a wide range of both financial and non-financial variables, but it was also a dataset with a lot of missing values. A large part of the missing values formed large clusters for some features and other missing values were randomly placed in the dataset. A combination of feature removal and data imputation techniques were found to be adequate to resolve these missing values.

We removed features that had over 50 percent missing data, the remaining features still contained some sparse missing values. These were resolved using linear interpolation with adjacent quarter values. If, for example, the ROE for Q3 2015 was absent, we would estimate it by taking the 2Q and 4Q values for 2015, averaging them, and assigning that as the ROE for Q3 2015.

Subsequently, the dataset was standardized to normalize the features. Each input variable was transformed to have a mean of 0 and a standard deviation of 1. This standardization ensures that all features contribute equally during model training. In addition to improving consistency, normalization enhances the model's convergence by reducing the risk of being trapped in local minima. Z-score normalization was used, defined by the following equation:

$$z = \frac{x - \mu}{\sigma} \quad (4.1)$$

Where:

μ is the mean of x

σ is the standard deviation of x

After completing the data preparation, the final dataset included 70 companies, each with 41 quarterly records over a 10-year period, resulting in a total of 2,091 records. For further analysis, 13 financial ratios were extracted, as detailed in Section 4.1.2.

4.1.4 Calculation of the Target Variable – Relative Return

The absolute returns quarterly of stock is defined as follows in Equation (4.2):

$$R_{absolute}(q) = \frac{cp(q)}{op(q)} - 1 \quad (4.2)$$

Where:

$R_{absolute}(q)$ is absolute returns quarterly of stock

$cp(q)$ is the stock's price on the last day of quarter q

$op(q)$ is the stock's price on the first day of quarter q .

To control movements in the market, and better isolate company specific performance, we measure the relative return by adjusting the absolute return of the stock with respect to a market benchmark. The relative quarterly return is defined in Equation (4.3):

$$R_{relative}(q) = R_{absolute}(q) - R_{TASI}(q) \quad (4.3)$$

$R_{absolute}$ is the quarterly absolute return of stock for quarter q , and the R_{TASI} is the quarterly return of the TASI as a benchmark portfolio. The absolute return of a stock is susceptible to influence not just from the company's performance but also from the broader stock market conditions. This inherent volatility makes it a less stable target for model learning. Consequently, we perform preprocessing by converting absolute returns into relative returns, which serves as the training target for our modeling efforts.

4.1.2. Data Split and Moving Time Period

The RF models encompass a total of 32 quarters for each moving period. The initial data from the moving period span from 2015Q1 to 2023Q1, and so forth. We partitioned these data into three segments: the training set, the validation set, and the test set, comprising 24 quarters, 8 quarters, and 1 quarter, respectively as in Figure 4.1. The training set facilitates model weight optimization, and the validation set aids in finetuning hyperparameters, ensuring model convergence without overfitting. Subsequently, we employ the test set to predict the relative return of the 51 stocks.

Year	2015				2016				2017				2018				2019				2020				2021				2022				2023				2024				2025			
Quarter	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Moving time period 1																																												
Moving time period 2																																												
Moving time period 3																																												
Moving time period 4																																												
Moving time period 5																																												
Moving time period 6																																												
Moving time period 7																																												
Moving time period 8																																												
Moving time period 9																																												

Figure 4.1. Rolling Window Data Splitting for Random Forest Training

The LSTM and GRU models span a total of 35 quarters for each moving period, with the initial data originating from 2015Q1 to 2023Q3 and so on. We segment this data into the training set, validation set, and test set, consisting of 20 quarters, 6 quarters, and 4 quarters, respectively as in Figure 4.2. Like the previous models, the training set facilitates model weight optimization, and the validation set aids in hyperparameter fine-tuning, guaranteeing model convergence without overfitting.

Year	2015				2016				2017				2018				2019				2020				2021				2022				2023				2024				2025			
Quarter	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4								
Moving time period 1																																												
Moving time period 2																																												
Moving time period 3																																												
Moving time period 4																																												
Moving time period 5																																												
Moving time period 6																																												
Moving time period 7																																												
Moving time period 8																																												
Moving time period 9																																												

Figure 4.2. Rolling Window Data Splitting for LSTM and GRU Training

4.1.4 Evaluation Metrics

To evaluate the effectiveness of the predictive models and the constructed portfolios, three key performance metrics were utilized:

4.1.4.1 Excess Return

This metric measures the difference between the average return of the constructed portfolio and that of the benchmark index (TASI), providing insight into the value added by the model's stock selection strategy.

Excess returns are a useful measure to evaluate the performance of an investment portfolio. By comparing portfolio returns to those of a benchmark, investors can determine whether the portfolio is generating value above the benchmark.

The excess return on a benchmark $ER_{TASI}(q)$ of portfolio p for quarter q is formulated as follows in Equation (4.4):

$$ER_p = \{\prod_{q=1}^n \mathbf{1} + R_p(q)\} - \{\prod_{q=1}^n \mathbf{1} + R_{TASI}(q)\} \quad (4.4)$$

Where:

$R_p(q)$ is the actual absolute return

$R_{TASI}(q)$ is the actual absolute return of TASI index

4.1.4.2 Top-k Precision

This assesses the accuracy of the model in identifying the true top-performing stocks within each quarter, offering a direct evaluation of predictive ranking quality. The precision of k is used to measure the percentage of overlap between a list of top k stocks with the highest predicted returns, which is ‘predicted top k stocks’, and a list of top k stocks with the highest actual return, which is ‘actual top k stocks’. The higher the precision of the top- k , the more accurate the prediction of the model will be. Top- k precision can be calculated as follows via Equation (4.5):

$$\text{Top} - k \text{ Precision} = \frac{|\text{Predicted } k \cap \text{Actual } k|}{k} \quad (4.5)$$

Where:

k The number of top stocks selected

$\cap$ Intersection, the number of stocks that appear in both the predicted and actual top-k lists

$$K \text{ Precision}_{\text{random}} = \frac{K}{\text{Total number of stocks}} \quad (4.6)$$

For Top-5 out of 51 stocks: $5/51 = 9.8\%$

For Top-10 out of 51stocks: $10/51 = 19.6\%$

4.1.4.3 Portfolio Score

A risk-adjusted measure that combines return and standard deviation, designed to capture both the magnitude and stability of investment performance over time. These metrics collectively offer a robust and comprehensive evaluation framework, capturing both predictive accuracy and real-world investment utility.

$$ER_p = \{\Pi_{q=1}^n \mathbf{1} + R_p(q)\} - \{\Pi_{q=1}^n \mathbf{1} + R_{TASI}(q)\} \quad (4.6)$$

4.1.3. Training

The training set, validation set, and test set are constructed by progressing through sequential time periods using a rolling window approach. Once a model is trained and its hyperparameters fine-tuned using the validation set, it is evaluated on the corresponding test set within that rolling period. In each test window, the model generates predicted relative returns for all 51 stocks in the investment universe. These predicted returns are then sorted in descending order to facilitate stock ranking.

Following this ranking process, two portfolios are formed: the Top-5 portfolio, composed of the five stocks with the highest predicted returns, and the Top-10 portfolio, containing the top ten predicted performers. This stock selection and portfolio formation process is repeated across all rolling periods, resulting in a dynamic, time-aware investment simulation, as shown in Figure 4.

Three distinct models were developed in this study: Random Forest (RF), implemented using the scikit-learn library in Python; and two recurrent neural networks, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), both implemented using the PyTorch deep learning framework.

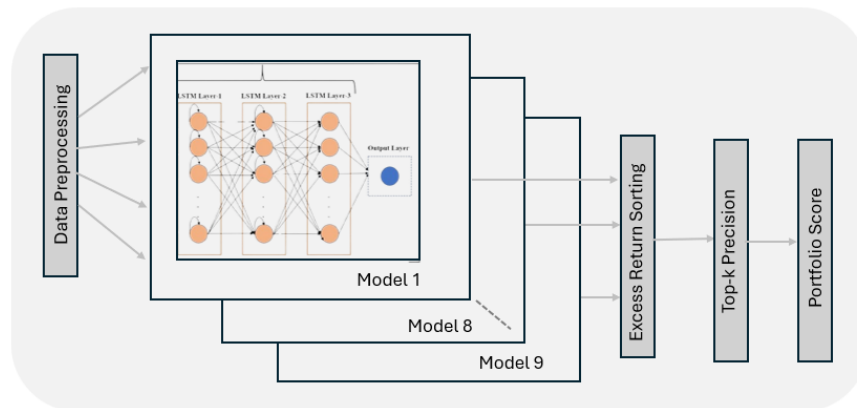


Figure 4.3 Model Training Architecture

4.1.3.1 Random Forest Hyperparameters

We use scikit-learn to develop the RF model. Hyperparameters are tuned with the validation set for preventing overfitting where the number of estimators is 1000 and the optimization criterion used SE.

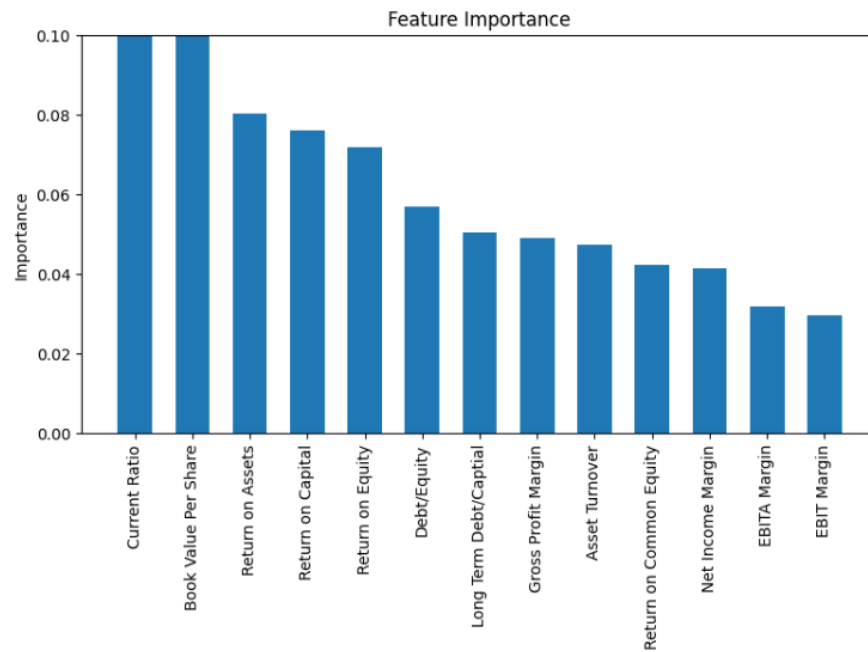


Figure 4.4. Importance of features of RF model

4.1.3.2 LSTM Hyperparameters

We use Pytorch to develop LSTM models. LSTM are tuned using the validation set to prevent overfitting, as shown in Table 4.2.

Name of Hyperparameters	Optimized Result
Hidden size	64
Epochs	100
Batch size	128
Dropout rate	0.4
Input size	13
Layers	2
Learning rate	0.0005
Optimizer	Adam

Table 4.2 Hyperparameters of LSTM model

4.1.3.3 GRU Hyperparameters

We use Pytorch to develop GRU models. Hyperparameters of GRU are tuned using the validation set to prevent overfitting, as shown in Table 4.3.

Name of Hyperparameters	Optimized Result
Hidden layers	1
Epochs	100
Batch size	128
Time step	12
Dropout rate	0.4
Input size	13
Layers	2
Learning rate	0.0005
Optimizer	Adam

Table 4.3 Hyperparameters of GRU model

Chapter 5 Experimental Results and Discussion

The evaluation compares three models RF, LSTM, and GRU using three key portfolio performance metrics: Portfolio Score, Excess Return relative to the TASI benchmark, and Top-k Precision for Top-5 and Top-10 stock selections.

Among the models, the GRU model achieved the strongest overall performance, attaining a Portfolio Score of 0.61 and an Excess Return of 6.4% in the Top-10 portfolio, outperforming both LSTM and RF. GRU also demonstrated robust stock selection ability, reaching a Top-10 Precision of 51%, reflecting its accuracy in identifying high-performing stocks.

The LSTM model followed closely, with a Top-10 Portfolio Score of 0.56, the highest Top-k Precision of 52%, and the highest Excess Return of 6.6%. This suggests that LSTM is particularly effective in ranking stocks accurately, even though its overall portfolio return is slightly below that of GRU.

By contrast, the Random Forest model showed the weakest results across all metrics, with Portfolio Scores of 0.54 (Top-5) and 0.55 (Top-10), Excess Returns of 4.7% and 5.3%, and Top-k Precisions of 33% and 42%, respectively. This indicates lower predictive power and stock selection capability compared to the deep learning models.

In summary, both LSTM and GRU significantly outperformed the TASI baseline (0% Excess Return), with GRU offering the most consistent and balanced performance. These findings highlight the advantages of recurrent neural networks in modeling temporal financial data for stock return prediction and portfolio-based investment strategies.

	RF		LSTM		GRU		TASI
Top - K	Top 5	Top 10	Top 5	Top 10	Top 5	Top 10	
Portfolio Score	0.54	0.55	0.55	0.56	0.58	0.61	
Excess return	4.7%	5.3%	6.1%	6.6%	5.8%	6.4%	Baseline 0%
Top-k Precision	33%	42%	40%	52%	39%	51%	

Table 5.1 Models Evaluation

For TASI Top-5 out of 51 stocks: $5/51 = 9.8\%$

For TASI Top-10 out of 51stocks: $10/51 = 19.6\%$

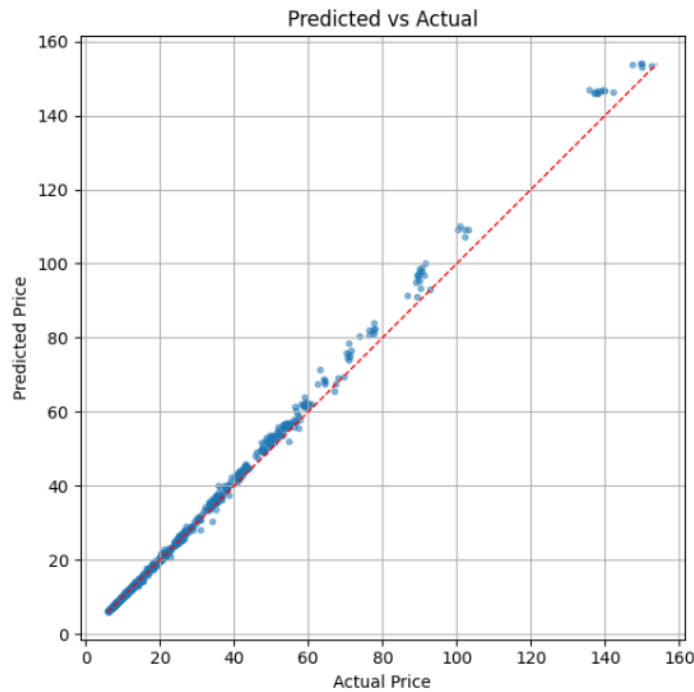


Figure 5.1 Predicted vs Actual Price

Using LSTM this scatter plot illustrates the relationship between the predicted stock prices and the actual stock prices in the test set. Each point represents a single prediction made by the model for a given stock in a specific quarter. The red dashed line corresponds to the ideal case where the predicted value exactly equals the actual value.

The data points are tightly clustered along the red line, indicating a strong correlation between predicted and actual prices. This suggests that the model demonstrates high predictive accuracy with minimal deviation across most price ranges. Only a few outliers deviate significantly from the line, especially at higher price levels, but overall, the model generalizes well across the full range of stock prices.

These results reflect the model's effectiveness in learning the underlying patterns in the training data and its ability to generalize to unseen stock prices in the test data.

Chapter 6 Conclusion and Future Work

This study explored the use of three predictive models RF, LSTM, GRU to forecast quarterly stock returns and construct investment portfolios. A rolling window framework was adopted to simulate a realistic investment scenario, where models are continually trained, validated, and tested across sequential time periods. This approach enabled the models to adapt to the evolving dynamics of the financial market.

The performance of each model was evaluated using Portfolio Score, Excess Return compared to the TASI index, and Top-k Precision. Results showed that deep learning models, particularly GRU and LSTM, provided improved predictive power and more consistent portfolio performance compared to the traditional RF model.

The GRU model, in particular, offered a strong balance between high excess returns and robust precision in selecting top-performing stocks.

Both Top-5 and Top-10 stock portfolios were constructed based on model predictions. While Top-5 portfolios showed potential for higher returns, Top-10 portfolios generally yielded more stable outcomes with reduced volatility. These findings underscore the importance of balancing return potential with risk exposure in portfolio strategy.

Looking ahead, future work could focus on enhancing model performance by incorporating additional predictive features such as macroeconomic indicators or sentiment data. There is also room to improve the optimization of model parameters and to explore alternative neural network architectures capable of capturing more complex temporal and cross-sectional patterns in financial data. Moreover, extending the evaluation framework to include real-world considerations like transaction costs, market liquidity, and risk-adjusted performance metrics would provide a more comprehensive assessment of practical investment outcomes.

References

1. Gupta, H., & Jaiswal, A. (2024). A study on stock forecasting using deep learning and statistical models. arXiv.
2. Tsai, P.-F., Gao, C.-H., & Yuan, S.-M. (2023). Stock selection using machine learning based on financial ratios. *Mathematics*, MDPI, 4758.
3. Lu, C., & Xu, Y. (2023). TRNN: An efficient time-series recurrent neural network for stock price prediction. *Information Sciences*, 640, 119–134.
4. Kunathananon, T. (2022). Estimating stock price based on information from financial statement using machine learning approach (Master's thesis, Chulalongkorn University). Chula Digital Collections.
5. Lu, Z.-Y. A Deep Reinforcement Learning-Enabled Portfolio Management System with Quarterly Stock Re-Selection Based on Financial Statements. Master's Thesis, National Yang Ming Chiao Tung University, Taiwan, China, 2022.
6. Huang, Y., Capretz, L. F., & Ho, D. (2021). Machine learning for stock prediction based on fundamental analysis. *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–10. Orlando, FL, USA.
7. Monahan, S. J. (2017). Financial statement analysis and earnings forecasting. *Foundations and Trends® in Accounting*, 12(2), 105–215.
8. Heo, J., & Yang, J. Y. (2016). Stock price prediction based on financial statements using SVM. *International Journal of Hybrid Information Technology*, 9(2), 57-66.
9. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
10. Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests Ensemble machine learning: Methods and applications (pp. 157–175). Springer.
11. Han, S., & Chen, R.-C. (2007). Using SVM with financial statement analysis for prediction of stocks. *Communications of the IIMA*, 7(4), Article 8.
12. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32

13. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780
14. GeeksforGeeks. (2025, July 23). Random forest algorithm in machine learning. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning>
15. Naveen. (2022, April 6). *What is the difference between LSTM and GRU?* NoMiDL. <https://www.nomidl.com/deep-learning/what-is-the-difference-between-lstm-and-gru>
16. Saudi Exchange. <https://www.saudiexchange.sa/>
17. Market Screener. <https://www.marketscreener.com/>
18. Investing.com. <https://www.investing.com/>
- 19.