國立臺灣大學共同教育中心統計碩士學位學程

碩士論文

Master Progarm of Statistics

Center for General Education

National Taiwan University

Master's Thesis

基於深度強化學習將動態市場條件納入投資組合最佳化

Incorporating Dynamic Market Conditions into Portfolio Optimization via Deep Reinforcement Learning

劉薇

Wei Liu

指導教授: 張智星, 蔡政安 博士

Advisor: Jyh-Shing Jang, Chen-An Tsai Ph.D.

中華民國 113 年 7 月

July, 2024

doi:10.6342/NTU202403155

國立臺灣大學碩士學位論文
口試委員會審定書
MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

基於深度強化學習將動態市場條件納入投資組合最佳化

Incorporating Dynamic Market Conditions into Portfolio Optimization via Deep Reinforcement Learning

本論文係 劉薇 (姓名) R11H41003 (學號) 在國立臺灣大學 統計 (系/所/學位學程)完成之碩士學位論文，於民國 113 年 7 月 24 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department / Institute of Statistics on 24 (date) 7 (month) 113 (year) have examined a Master's thesis entitled above presented by Wei Liu (name) R11H41003 (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

張智星 (指導教授 Advisor)　　蔡政安

陳[illegible]　　韓傳祥

系主任/所長 Director: 共同教育中心統計碩士學位學程主任 蔡政安

doi:10.6342/NTU202403155

Acknowledgements

回顧這兩年的碩士生活，最感謝的就是指導教授張智星老師，除了提供非常多的研究資源及新資訊外，每次的會議老師總能一語破的指出了我在研究上的缺失及報告內容上的不足，透過每次的修正也讓我在完成論文的同時，學習到許多研究方法和表達技巧。此外，我要感謝陳永耀老師和博士班學長莊謹譽。在每週的分組會議中對我的研究提供了寶貴的反饋和討論意見。從一開始的計劃構思，到不同主題的論文閱讀、測試和改進，再到最後的定題和深入探討，您們的幫助對我意義重大。另外也要感謝統計學程的蔡政安主任，不僅給予學程學生權力，讓我有機會到感興趣的研究領域跟隨不同科系的老師進行研究，還提供了建設性的建議，提升了我論文的完整性。同時也要感謝韓傳祥老師，從大學時期對我在量化交易上的一席教導令我受益匪淺，能在碩士論文階段再次得到您的指導實在是倍感榮幸。

最後要謝謝統計學程辦公室的所有助理，對於學生的需求總是盡力達成，謝謝 MIRLAB 一起奮鬥的同學們及王崇喆學長，協助國網中心的額度設置，也在口試預演時提出許多建議，謝謝我的父母，在求學過程中讓我無後顧之憂，去做一切想做的嘗試，謝謝我的朋友們，陪我一起解決問題，共同面對挑戰，讓這兩年研究生活充實圓滿。

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

摘要

本研究探討在動態且複雜的金融市場中優化投資組合策略的挑戰。傳統方法如 Markowitz 均值-方差模型，由於其靜態性和對歷史數據的依賴，在實際應用中往往效果不佳。深度強化學習（Deep Reinforcement Learning）模型，特別是 DeepTrader，已顯示出適應市場變化的潛力。然而，這些模型在應用於不同市場（如台灣市場）時，往往會出現極端表現變化的問題。本研究旨在通過整合 Transformer 網絡來改進 DeepTrader 模型，替換其原始模型中的 Graph Convolutional Network 和 Long Short-Term Memory 部分。Transformer 網絡因其優越的長期依賴關係捕捉能力和對股票間複雜關聯的處理能力而被選中。我們在美國和台灣市場上測試了所提出的改進模型，以評估其性能和穩定性。具體實驗包括在美國市場上運用 DeepTrader 模型，深入分析其在不同市場條件下的行為和性能表現，並比較 DeepTrader 模型在台灣市場上的表現，識別其在長期關聯學習中的不足之處。此外，我們測試改進後的 Transformer 模型，評估其在穩定性和適應不同市場環境方面的提升。研究結果表明，改進後的模型在處理市場波動和適應不同市場環境方面顯著優於原始 DeepTrader 模型，特別是在應對市場變化和捕捉長期依賴關係方面顯示出更強的能力。每月更新權重又可以將改進後的 Transformer 模型效果發揮得更好，使其在動態市場環境中保持更高的穩定性和準確性。

關鍵字：投資組合優化、深度強化式學習、Transformer

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

Abstract

This research addresses the challenges of optimizing portfolio strategies in dynamic financial markets. Traditional methods like the Markowitz mean-variance model often fail due to their static nature. Deep reinforcement learning (DRL) models, particularly DeepTrader, show promise but struggle with performance variability across different markets, such as Taiwan. We aim to enhance DeepTrader by integrating Transformer networks, replacing the Graph Convolutional Network and Long Short-Term Memory components. Transformers were chosen for their ability to capture long-term dependencies and handle complex stock relationships. We tested the improved model in both U.S. and Taiwanese markets. Key findings include analyzing DeepTrader's behavior in the U.S. market under various conditions, identifying performance issues in the Taiwanese market related to long-term correlation learning, and demonstrating that the Transformer-based model improves stability and adaptability across different markets. The improved model significantly outperforms the original DeepTrader in handling market volatility and adapting to

doi:10.6342/NTU202403155

diverse market environments, showing stronger capabilities in capturing long-term dependencies and responding to market changes. Additionally, implementing monthly updates to the model's weights further enhances the performance of the Transformer-based model, ensuring greater stability and accuracy in dynamic market environments.

Keywords: Portfolio optimization, Deep reinforcement learning, Transformer

doi:10.6342/NTU202403155

Contents

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

List of Figures

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

List of Tables

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

Chapter 1 Introduction

This chapter provides an overview of the study, highlighting the motivation behind the research, the challenges faced in current financial modeling approaches, and the objectives and contributions of this work. It sets the stage for the detailed investigation and analysis conducted in the subsequent chapters.

1.1 Background

The dynamic and complex nature of financial markets necessitates efficient investment strategies to cope with market changes. Traditional portfolio optimization methods, such as the Markowitz mean-variance model [4], typically assume static market conditions and rely on historical data, often leading to sub-optimal performance in real-world applications. As market conditions become increasingly volatile and unpredictable, these traditional algorithms often fall short. Consequently, there has been a shift towards learning-based methods to address these challenges.

Reinforcement Learning (RL), especially Deep Reinforcement Learning (DRL), has gained widespread application in financial markets in recent years due to its ability to learn optimal strategies in dynamic environments. Reinforcement learning algorithms such as Q-learning, Deep Q-Networks (DQN) [5], and policy gradient methods have been applied

doi:10.6342/NTU202403155

to stock trading, futures trading [6], and other financial domains [7]. These algorithms are designed to learn and adapt continuously, making them well-suited for the constantly evolving nature of financial markets.

In 2017, JPMorgan introduced LOXM [8], an advanced RL-based trading program for real-time trading, which marked one of the earliest applications of RL in high-frequency trading by a major financial institution. This trend has continued, with leading hedge funds such as Bridgewater Associates and Two Sigma increasingly adopting RL as their preferred trading strategy [9]. Additionally, the market has seen the emergence of funds entirely managed by RL-generated strategies, which have demonstrated impressive performance.

The primary reason for choosing RL in financial markets is its learning methodology, which is analogous to the human brain's reward-based learning but without human emotional biases. This capability allows RL algorithms to make real-time adjustments to models based on market conditions, enhancing the robustness and adaptability of trading strategies. By minimizing human emotional interference and responding dynamically to market changes, RL provides a promising approach for developing more effective and resilient investment strategies.

1.2 Research Problem

Many current deep reinforcement learning models have achieved significant success in specific markets like the U.S. market [10] [11] [3]. These models have demonstrated the ability to generate profitable trading strategies, optimize portfolios, and manage risks effectively within the relatively stable and well-studied context of the U.S. financial mar-

doi:10.6342/NTU202403155

ket. However, when these models are applied to different markets, such as the Taiwanese market, they encounter significant challenges.

One of the primary issues observed is the extreme volatility in strategy performance. This volatility manifests as large swings in returns, which can lead to substantial gains or losses in a short period. Such behavior is problematic for investors seeking consistent and reliable returns. The underlying causes of this volatility include differences in market structure, trading volume, investor behavior, and regulatory environment between the U.S. and Taiwanese markets. These factors can affect how financial models interpret data and make predictions, leading to less reliable performance in new contexts.

1.3 Research Objectives

This paper aims to address the limitations of the DeepTrader (DT) model by enhancing its adaptability and performance in the Taiwanese market. The primary objective is to improve the robustness and generalization capabilities of the model to ensure that it performs effectively across different market environments. To achieve this, we will introduce Transformers, a state-of-the-art machine learning architecture known for its superior handling of sequential data and long-term dependencies [12]. The Transformers will be used to replace parts of the Graph Convolutional Network (GCN) and Long Short-Term Memory (LSTM) components in the DeepTrader model.

Specifically, we aim to:

1. Enhance Model Architecture: By incorporating Transformer technology, we seek to improve the model's ability to capture complex temporal and spatial relationships

doi:10.6342/NTU202403155

in financial data, which are critical for making accurate predictions and decisions in dynamic markets [13].

2. Cross-Market Evaluation: Evaluate the modified model's performance using extensive datasets from both the U.S. and Taiwanese markets. This involves training and testing the model on historical market data, followed by a comparative analysis to assess improvements.

3. Parameter Tuning and Optimization: Explore different configurations of the Transformer-based model to identify the optimal settings that yield the best performance in the Taiwanese market.

1.4 Research Contributions

The contributions of this work are multifaceted and significant, providing both theoretical advancements and practical insights into the application of deep reinforcement learning in financial markets.

1. Proposing an Improved Portfolio Optimization Model: This work introduces a novel portfolio optimization model that incorporates Transformer technology. By leveraging the Transformer's ability to capture complex data relationships and handle market noise, this model aims to significantly enhance predictive accuracy and decision-making capabilities.

2. Conducting a Detailed Comparative Analysis: A comprehensive analysis of the model's performance in both the U.S. and Taiwanese markets will be conducted. This analysis aims to evaluate the model's strengths and weaknesses across differ-

doi:10.6342/NTU202403155

ent market conditions, using various performance metrics to provide a thorough assessment of its risk and return characteristics.

3. Providing Insights into Model Adaptability and Applicability: The study will explore the adaptability and applicability of advanced deep reinforcement learning models in diverse financial environments. By examining the model's response to different market dynamics, such as liquidity and volatility, the study aims to identify factors influencing performance and guide future research and development in financial modeling.

By addressing these objectives, this study aims to contribute to the field of financial market analysis by enhancing the robustness and versatility of investment strategies through advanced machine learning techniques. The proposed improvements are expected to bridge the performance gap between different market environments, making DRL-based trading models more reliable and widely applicable.

1.5 Chapter Overview

This chapter offers an overview of our study, outlining its purpose and significance. In Chapter 2, we explore the background literature and review existing research relevant to our work. Chapter 3 details the methodology employed in our study, explaining the techniques and processes used. The experimental setup, including data collection, evaluation criteria, and parameter settings, is presented in Chapter 4. Chapter 5 provides an in-depth analysis of the experimental results, discussing their implications and relevance. The final chapter, Chapter 6, wraps up the study by summarizing the main conclusions, addressing the study's limitations, and proposing avenues for future research.

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

Chapter 2 Related Work

This chapter reviews the existing literature and methodologies relevant to our study. It discusses the advancements and limitations of Graph Convolutional Networks (GCNs) [1], Long Short-Term Memory (LSTM) networks [14], Reinforcement Learning (RL) frameworks, and Transformer architectures [12] in financial market analysis. The chapter provides a foundation for understanding the rationale behind integrating these methodologies into the DeepTrader model.

2.1 Graph Convolutional Network

Graph Convolutional Networks (GCNs) have been increasingly used in financial market analysis to capture the intricate relationships between different financial entities. GCNs apply convolution operations on graphs, where nodes represent entities (e.g., stocks) and edges represent relationships (e.g., correlations). In financial markets, GCNs can model the connectivity and influence between different stocks, sectors, or even broader market indices. For instance, by representing a financial market as a graph where nodes are stocks and edges are their correlations, GCNs can learn patterns that influence stock movements [15] [16]. This capability makes GCNs particularly useful for portfolio optimization and risk management, as they can effectively capture the interdependencies

doi:10.6342/NTU202403155

within a market. However, GCNs also have limitations, especially in capturing long-term dependencies and temporal dynamics, which is why combining them with LSTMs or replacing parts with Transformers can be beneficial. The Figure 2.1 illustrates the graph structure of GCN.

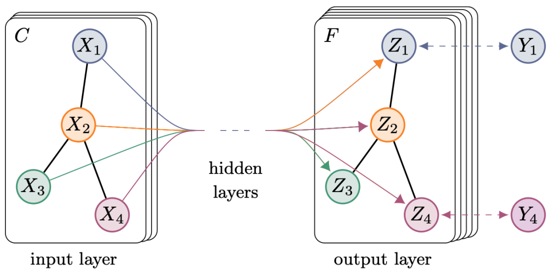

Figure 2.1: Graph Convolutional Network [1]

2.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) that are particularly well-suited for handling and predicting time series data. LSTMs have been widely applied in financial time series forecasting, such as stock price prediction and trading signal generation [17] [18]. The Figure 2.2 illustrates the detailed architecture of LSTM model.

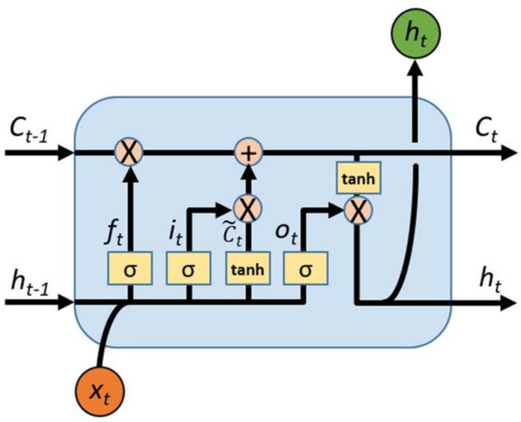

Figure 2.2: Model architecture of LSTM [2]

doi:10.6342/NTU202403155

LSTM networks excel in capturing long-term dependencies in sequential data, making them ideal for financial market applications where historical data and trends significantly influence future movements. Specific applications include single-step predictions, where the model predicts the next value in a sequence, and multi-step predictions, where the model forecasts a series of future values. These capabilities make LSTMs valuable for tasks such as stock price forecasting and developing trading strategies that depend on historical price movements and patterns [19].

2.3 Reinforcement Learning

Reinforcement Learning (RL) has gained significant traction in financial markets due to its ability to learn optimal strategies in dynamic environments. RL's flexibility and adaptive nature make it particularly well-suited for the ever-changing conditions of financial markets. This section will explore several notable RL frameworks and models that have been applied to portfolio management and trading strategies [20] [21] [22].

1. Ensemble of Identical Independent Evaluators (EIIE) [10]: One notable RL framework is the Ensemble of Identical Independent Evaluators (EIIE), which utilizes RL for portfolio management. The EIIE framework involves training an RL agent to make investment decisions that maximize returns while managing risks. The agent learns by interacting with the environment, receiving feedback in the form of rewards or penalties based on its actions. This feedback loop allows the agent to refine its strategy over time. The EIIE framework has demonstrated the effectiveness of RL in balancing the trade-off between risk and return, adapting to market fluctuations, and making informed investment decisions.

doi:10.6342/NTU202403155

2. Alphastock Model [11]: Another prominent RL model designed for stock trading is Alphastock. Alphastock uses a combination of LSTM and an attention mechanism to predict stock prices and generate trading signals. The model leverages the temporal dependencies of stock prices through LSTM, while the attention mechanism helps focus on relevant features for accurate predictions. This combination allows Alphastock to capture intricate patterns in historical price data and make informed trading decisions. By integrating RL techniques, Alphastock continuously improves its trading strategy, enhancing its ability to navigate the complexities of stock markets.

3. DeepTrader Model [3]: The DeepTrader (DT) model is another significant application of RL in financial markets. DeepTrader employs a multi-agent reinforcement learning approach to optimize trading strategies. Each agent in the model is responsible for managing a specific aspect of the trading process, such as asset allocation, risk management, or order execution. The agents interact with the market environment, receiving rewards based on their performance, and collectively learn to make optimal trading decisions. DeepTrader's multi-agent framework enables a comprehensive approach to portfolio management, accounting for various market factors and enhancing overall performance.

   The DeepTrader model integrates multiple advanced machine learning components to optimize trading strategies. The Figure 2.3 illustrates the detailed architecture of the DeepTrader model and the architecture consists of the following key units:

   (a) **Asset Scoring Unit (ASU)**: This unit is responsible for processing input data to learn asset-specific representations. It uses a combination of a Temporal Convolutional Network (TCN) and a Spatial Attention mechanism to capture

doi:10.6342/NTU202403155

both temporal and spatial dependencies in the data. The ASU includes residual connections to stabilize the learning process. The output from the ASU is passed to the Portfolio Generator.

(b) **Market Scoring Unit (MSU)**: This unit processes market-wide data using a Transformer with Temporal Attention to capture market dynamics. The MSU's output provides a comprehensive understanding of the overall market conditions.

(c) **Portfolio Generator**: This component takes the outputs from both the ASU and MSU to generate portfolio weights (ω^+, ω^-) and a risk parameter (ρ). The Portfolio Generator utilizes fully connected layers to synthesize the learned representations and make final portfolio decisions.

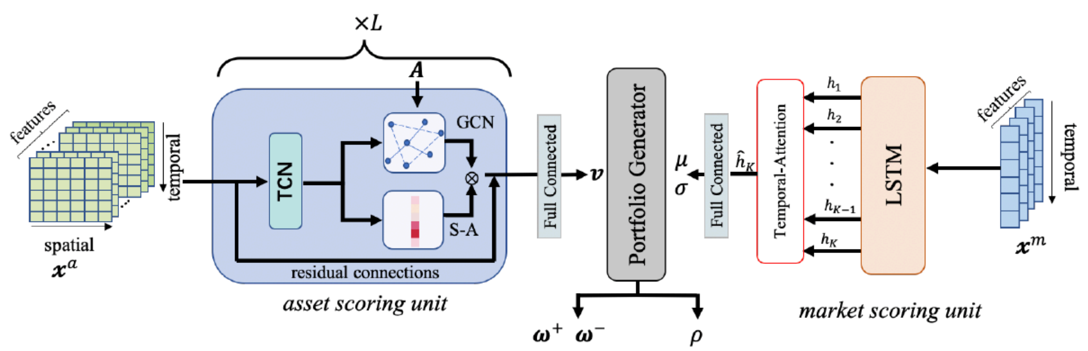

Figure 2.3: Model architecture of DeepTrader [3]

The application of RL in financial markets has shown promising results, with RL models outperforming traditional approaches in various scenarios. These models leverage RL's ability to learn from experience, adapt to changing market conditions, and make informed decisions based on historical data. By continuously refining their strategies through trial and error, RL models have the potential to improve portfolio management and trading performance, offering valuable insights for investors and financial professionals.

doi:10.6342/NTU202403155

2.4 Transformer

Transformer, originally developed for natural language processing tasks, have shown great potential in handling sequential data and capturing long-range dependencies. Their self-attention mechanism allows them to weigh the importance of different parts of a sequence, making them suitable for financial market analysis. Transformers can model complex temporal patterns and relationships in financial data, leading to more accurate predictions and better trading strategies. Recent studies have applied Transformers to stock price prediction, volatility forecasting, and portfolio optimization with promising results [13] [23] [24].

This paper proposes integrating Transformers into the DeepTrader model to enhance its performance and adaptability in different market conditions, such as the Taiwanese market. By leveraging the Transformer's capabilities, we aim to address the limitations of GCNs and LSTMs and improve the overall robustness and effectiveness of the model.

2.5 Summary

While traditional GCNs and LSTMs have demonstrated success in financial market analysis, their limitations in capturing long-term dependencies and temporal dynamics can hinder their performance in volatile and diverse market environments. Reinforcement learning models like EIIE, Alphastock and DeepTrader have shown promise in portfolio management and trading strategies, but there is room for improvement in their adaptability and robustness across different markets. By integrating Transformers into these models, we aim to enhance their capabilities and provide more reliable and effective solutions for

doi:10.6342/NTU202403155

financial market analysis and trading, particularly for the Taiwanese market.

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

Chapter 3 Method

This chapter presents the methodologies used in this research to enhance financial trading models. It begins with an exploration of optimization techniques based on the DeepTrader framework, providing a foundation for understanding its application in financial settings. The chapter then delves into Transformer-based attention mechanisms, highlighting their role in improving model performance. Detailed descriptions of model components are provided to elucidate how these elements contribute to the overall model architecture.

3.1 Optimization Based on DeepTrader

One of the key advantages of the DeepTrader model is its ability to separate the training of assets and markets. This modular approach allows DeepTrader to capture the unique characteristics of individual assets while also understanding broader market dynamics. By leveraging Graph Convolutional Networks (GCNs) and Long Short-Term Memory (LSTM) networks, DeepTrader can effectively model the relationships between different assets and their temporal price movements.

However, DeepTrader also has its drawbacks. One major limitation of using GCNs is that they typically require a predefined graph structure to handle heterogeneous data,

doi:10.6342/NTU202403155

such as prices and technical indicators. In dynamic and complex market environments, this predefined structure can be restrictive and may not fully capture the intricacies of the data. Additionally, GCNs can be computationally inefficient when dealing with large-scale data. GCNs rely on the local neighborhood structure of the graph, which can be computationally expensive to process, especially when the market data is extensive and frequently changing. This can lead to significant computational costs and time delays, making GCNs less suitable for real-time applications in large and volatile markets.

Moreover, DeepTrader has shown limited effectiveness in learning market sentiment and broader macroeconomic factors, which are crucial for comprehensive trading strategies. While the combination of GCN and LSTM in the DeepTrader model has been effective in many scenarios, it presents several limitations that restrict its adaptability and performance in dynamic market environments. These limitations are primarily due to the inherent characteristics and operational constraints of GCNs and LSTMs.

The predefined graph structure in GCNs is static and typically based on historical data, which may not adapt to real-time changes in market relationships influenced by macroeconomic events, geopolitical developments, and sudden market sentiments. This lack of adaptability can lead to potential inaccuracies in modeling asset relationships and may not effectively capture the complex, non-linear relationships in financial markets.

Computational inefficiency is another significant drawback. GCNs involve convolution operations on graphs, which can be computationally intensive. As the size of the graph increases, the computational cost grows significantly. In real-time trading applications, timely decision-making is crucial, and the computational burden of processing large-scale graphs can lead to delays, making the model less effective for high-frequency

doi:10.6342/NTU202403155

trading or real-time strategy adjustments. The high computational resources required can also be a barrier for individual traders or smaller financial institutions with limited computing infrastructure.

Furthermore, the GCN and LSTM combination in DeepTrader primarily focuses on structural relationships between assets and temporal dynamics of individual stock prices. This approach often overlooks broader market sentiment and macroeconomic factors that can significantly impact asset prices. Market sentiment, driven by news, social media, and investor psychology, plays a critical role in financial markets. Traditional GCNs and LSTMs do not inherently capture these sentiment signals unless explicitly integrated into the model, which adds complexity and requires additional data sources. Additionally, broader economic indicators such as interest rates, employment figures, and geopolitical events influence market dynamics, but the original DeepTrader model's architecture does not explicitly account for these macroeconomic variables, potentially limiting its predictive accuracy and robustness.

The implications of these limitations are significant. The inability to adapt to real-time changes and capture complex, evolving relationships can lead to suboptimal trading strategies, especially in volatile or less predictable markets like the Taiwanese market. High computational demands restrict the model's applicability in real-time trading scenarios and make it less accessible to entities with limited computational resources. Focusing primarily on structural and temporal data without integrating sentiment and macroeconomic factors results in an incomplete representation of market dynamics, which can impair the model's decision-making process.

doi:10.6342/NTU202403155

3.2 Transformer-based Attention

To address these limitations, the introduction of Transformer models offers a more flexible and efficient solution. Transformers, with their advanced self-attention mechanisms, can dynamically learn relationships and dependencies without relying on predefined structures, thus enhancing adaptability to changing market conditions. They also process data more efficiently, making them suitable for real-time applications, and can integrate a broader range of market signals, including sentiment and macroeconomic factors, providing a more holistic view of the market.

By incorporating Transformers into the DeepTrader framework, the model can overcome these limitations, resulting in more robust and effective trading strategies across diverse market environments.

3.2.1 Enhancements with Transformer Models

Transformers bring several key advantages to the DeepTrader model, particularly through their self-attention mechanisms, which allow for better handling of long-term dependencies and complex relationships between stocks.

1. **Transformer Model Basics**

    - **Self-Attention Mechanism**: This allows the model to weigh the importance of different parts of the input data, enabling it to focus on relevant information while ignoring irrelevant details.

    - **Multi-Head Attention**: This extends the self-attention mechanism by allowing the model to attend to information from multiple representation subspaces

doi:10.6342/NTU202403155

at different positions.

- **Positional Encoding**: Since Transformers do not have a built-in notion of the sequential order of the input data, positional encoding is added to provide information about the position of each element in the sequence.

2. **Rationale for Choosing Transformers**

Transformers are chosen to replace GCN and LSTM due to their ability to better capture long-term dependencies and complex relationships between stocks. Unlike RNNs, which process data sequentially, Transformers can process entire sequences simultaneously, leading to more efficient learning. Furthermore, the self-attention mechanism can model intricate dependencies between different stocks and their historical data, potentially improving the model's ability to adapt to dynamic market conditions.

3. **Improved Model Architecture**

The proposed architecture includes:

- **Input Layer**: Processes raw financial data, including historical prices, technical indicators, and macroeconomic variables.
- **Transformer Encoder**: Replaces the GCN and LSTM components to capture both the interrelationships between stocks and the temporal dynamics of the market.
- **Output Layer**: Generates the optimized portfolio strategy based on the learned representations.

doi:10.6342/NTU202403155

3.2.2 Model Architecture

The model architecture, as illustrated in Figure 3.1, integrates Transformer models to enhance the overall effectiveness and adaptability of the trading strategies. The following components are key to this architecture:

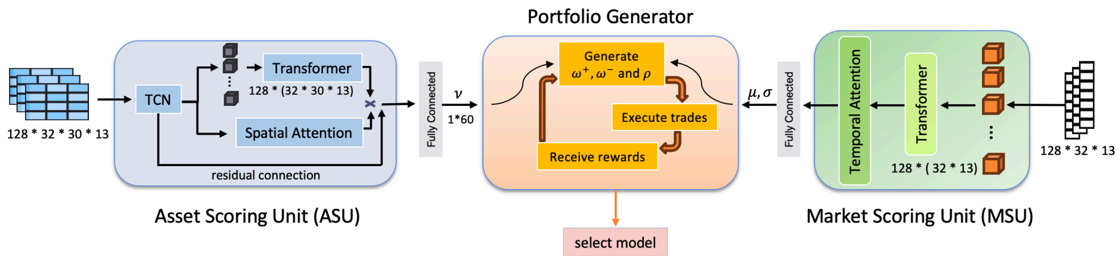

Figure 3.1: Model architecture of new DeepTrader

The model architecture consists of three main components: the Asset Scoring Unit (ASU), the Portfolio Generator (PG), and the Market Scoring Unit (MSU). This section provides a detailed explanation of each component and their interactions.

3.2.2.1 Asset Scoring Unit (ASU)

The Asset Scoring Unit (ASU) is responsible for analyzing individual assets and generating asset scores that reflect their potential for investment.

1. **Input Data:** The input to the ASU is a tensor of shape $128 \times 32 \times 30 \times 13$, where 128 represents the size of features, 32 is the batch size, 30 is the number of assets, and 13 is the number of time steps. Initially, the input features have a length of 34, which is then transformed into a length of 128 through an embedding layer.

2. **Temporal Convolutional Network (TCN):** The input data first passes through a Temporal Convolutional Network (TCN) to capture temporal dependencies.

doi:10.6342/NTU202403155

3. **Data Flattening:** After the TCN, the data is flattened to prepare it for the Transformer. This process reshapes the tensor into a format suitable for the subsequent layers.

4. **Transformer:** The flattened data is fed into a Transformer network, which processes the sequential data to capture long-term dependencies and complex patterns.

5. **Spatial Attention:** A Spatial Attention mechanism is employed to focus on relevant features across different dimensions, enhancing the model's ability to identify important signals.

6. **Residual Connection:** The residual connection ensures the stability of the learning process.

7. **Output:** The output from the ASU is a vector **v** of shape 1×60, representing the scores for 60 assets, which is then fully connected and passed to the Portfolio Generator.

3.2.2.2 Market Scoring Unit (MSU)

The Market Scoring Unit (MSU) evaluates the overall market conditions and generates market scores that guide the portfolio generator.

1. **Input Data:** The input to the MSU is a tensor of shape $128 \times 32 \times 13$, representing 128 time steps, each with 32 features and 13 channels.

2. **Data Flattening:** Similar to the ASU, the input data is flattened before being processed by the Transformer. This step reshapes the tensor into a format compatible with the following layers.

doi:10.6342/NTU202403155

3. **Transformer:** The flattened data is fed into a Transformer network, which processes the sequential data to capture long-term dependencies and complex patterns.

4. **Temporal Attention:** A Temporal Attention mechanism is employed to focus on relevant time steps, improving the model's ability to identify important temporal signals.

5. **Output:** The output from the MSU generates a risk parameter (ρ), which is used by the Portfolio Generator to adjust trading strategies according to the current market conditions.

3.2.2.3 Portfolio Generator

The Portfolio Generator (PG) component is designed to optimize trading strategies using reinforcement learning (RL). The core idea is to generate optimal asset weights (ω^+, ω^-) and a risk aversion parameter (ρ) that maximize the portfolio's performance.

- **Reinforcement Learning Framework:** In the RL framework, the portfolio generator acts as an agent that interacts with the market environment. The agent receives states from the environment, which consist of various financial indicators and historical data. Based on these states, the agent takes actions, which involve generating the portfolio weights and the risk parameter.

- **Reward Function:** The reward function plays a crucial role in guiding the learning process of the RL agent. In this setup, the reward function is designed to maximize the annual return (R_{annual}). The annual return is calculated as:

$$R_{\text{annual}} = \left(\frac{V_{t+T}}{V_t} \right)^{\frac{1}{T}} - 1$$

doi:10.6342/NTU202403155

where V_t is the portfolio value at time t and T is the investment horizon.

- **Model Selection:** After training the RL agent, the final model selection is based on the Calmar ratio. The Calmar ratio is a performance measure that considers both the annual return and the maximum drawdown (MDD). It is defined as:

$$\text{Calmar Ratio} = \frac{R_{\text{annual}}}{MDD}$$

  where the maximum drawdown is the largest peak-to-trough decline in the portfolio value over a specified period. This metric ensures that the selected model not only achieves high returns but also maintains a controlled level of risk.

By integrating the Transformer models with attention mechanisms into both the ASU and MSU, the improved architecture can dynamically learn and adapt to the evolving market conditions, enhancing the overall robustness and effectiveness of the trading strategies.

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

Chapter 4 Experimental Setup

This chapter outlines the experimental setup for evaluating the proposed models. It begins by describing the datasets used for testing, including market data from the US and Taiwan. The chapter details the evaluation metrics employed to assess model performance, followed by an explanation of the experimental environment and parameter settings. The chapter concludes with a roadmap of experiments, specifying the objectives and procedures for each test, which include analyzing market interactions, testing in different markets, improving models with Transformers, and retraining with updated data.

4.1 Dataset

The dataset used in this study encompasses price information, technical indicators, and macroeconomic data from both the US and Taiwanese markets. Sourced from financial databases and APIs, the data undergoes cleaning, normalization, and feature extraction to prepare it for modeling. Technical indicators include common measures such as Moving Averages (MA20 and MA60), Relative Strength Index (RSI), Stochastic Oscillator (K/D), Moving Average Convergence Divergence (MACD), and Bollinger Bands (BBANDS). Additionally, Technical indicators include Alpha factors which are employed to evaluate and predict stock performance, aiding in the construction and refinement of trading

doi:10.6342/NTU202403155

strategies and portfolio optimization techniques.

Alpha factors aim to capture various aspects of market behavior and stock performance. They serve several purposes including momentum analysis, volatility measurement, volume analysis, trend detection, risk and return assessment, and correlation and beta analysis. These factors are used to construct and refine trading strategies and portfolio optimization techniques in financial models.

In this paper, the Alpha factors derived from Alpha101 [25] and used include Alpha001, Alpha002, Alpha003, Alpha012, Alpha019, Alpha025, Alpha033, Alpha038, Alpha040, Alpha44, Alpha45, Alpha46, Alpha51, Alpha52, Alpha53, Alpha54, Alpha55, Alpha56, Alpha60, Alpha62, Alpha068, Alpha075, Alpha83 and Alpha101. These factors provide insights into stock performance and aid in developing sophisticated trading and investment strategies.

4.1.1 US Market

The US market dataset includes DJIA component prices, technical indicators, and macroeconomic data as summarized in Table 4.1. The data coverage period is from 1992/01/01 to 2024/02/29.

4.1.2 TW Market

The TW market dataset includes 0050.TW component prices, technical indicators, and macroeconomic data as summarized in Table 4.2. The data coverage period is from 2000/01/01 to 2024/02/29.

doi:10.6342/NTU202403155

<table>
<tr><th>Unit</th><th colspan="2">Features</th></tr>
<tr><td rowspan="2">ASU (Asset Scoring Unit)</td><td colspan="2">Price information of components of DJIA (total=30)</td></tr>
<tr><td colspan="2">Technical indicators (a total of 34 kinds)</td></tr>
<tr><td rowspan="9">MSU (Market Scoring Unit)</td><td rowspan="3">Stock index</td><td>S&P500</td></tr>
<tr><td>DJIA</td></tr>
<tr><td>VIX</td></tr>
<tr><td rowspan="2">Government bonds</td><td>10-year</td></tr>
<tr><td>30-year</td></tr>
<tr><td rowspan="3">Corporate bonds</td><td>Credit rating: AAA</td></tr>
<tr><td>Credit rating: BBB</td></tr>
<tr><td>Credit rating: CCC</td></tr>
<tr><td>Exchange</td><td>XAU/USD</td></tr>
</table>

Table 4.1: Summary of ASU and MSU Components in the US market

<table>
<tr><th>Unit</th><th colspan="2">Features</th></tr>
<tr><td rowspan="2">ASU (Asset Scoring Unit)</td><td colspan="2">Price information of components of 0050.TW (total=27)</td></tr>
<tr><td colspan="2">Technical indicators (a total of 34 kinds)</td></tr>
<tr><td rowspan="6">MSU (Market Scoring Unit)</td><td>Stock index</td><td>TAIEX</td></tr>
<tr><td rowspan="4">Government bonds</td><td>5-year</td></tr>
<tr><td>10-year</td></tr>
<tr><td>20-year</td></tr>
<tr><td>30-year</td></tr>
<tr><td>Exchange</td><td>TWD/USD</td></tr>
</table>

Table 4.2: Summary of ASU and MSU Components in the TW market

4.2 Evaluation Metrics

In evaluating financial models, we primarily use the following four metrics: annual return, Sharpe ratio, maximum drawdown, and winning rate relative to the market. These metrics provide a comprehensive assessment of model performance and risk characteristics. The details are as follows:

- **Annual return**: This is a key metric for measuring the long-term performance of an investment strategy. It reflects the average return rate of the portfolio over a year and provides a direct measure of the strategy's profitability. By comparing annualized returns of different strategies, we can determine which strategy yields higher returns.

doi:10.6342/NTU202403155

- **Sharpe ratio**: This metric measures the excess return per unit of risk. The Sharpe ratio is calculated by subtracting the risk-free rate from the annualized return and then dividing by the annualized volatility. A higher Sharpe ratio indicates that the strategy provides better returns for the level of risk taken. It helps us compare risk-adjusted returns across different strategies and choose the optimal investment plan.

- **Maximum drawdown**: Maximum drawdown assesses the risk of an investment strategy by measuring the largest peak-to-trough decline in the portfolio's value. It helps us understand the strategy's performance during market downturns and evaluate its risk management capabilities. A smaller maximum drawdown typically indicates better risk control, making the strategy more suitable for risk-averse investors.

- **Winning rate**: The winning rate is the proportion of winning trades in the strategy. The winning rate relative to the market helps us understand the frequency of successful trades compared to market performance. This metric assists investors in determining the proportion of capital to allocate and filtering out strategies that experience extreme gains or losses. A higher winning rate suggests that the strategy consistently achieves positive returns, increasing investor confidence and decision quality.

By using these metrics, we can evaluate the overall return and risk of strategies and understand their practical effectiveness and market adaptability. These evaluation standards aid in making more informed decisions among various strategies.

doi:10.6342/NTU202403155

4.3 Environment

All experiments, including both training and test, are conducted on TWCC. The configuration for TWCC is as follows:

- **CPU**: Intel(R) Xeon(R) Platinum 8280 @ 2.70 GHz
- **RAM**: 90 GB
- **GPU**: NVIDIA Tesla V100 SXM2

4.4 Parameter Settings

During training, we use a learning rate of 1×10^{-6}. The model has a hidden dimension of 128 and consists of 4 blocks, with a kernel size of 2. Dropout is applied with a rate of 0.5 to prevent overfitting. We utilize the Adam optimizer with the specified learning rate.

The training is conducted with a batch size of 32 and a window length of 13. The model is trained for 1500 epochs, with a maximum gradient norm of 100.0 and weight decay set to 0.001 for regularization. Momentum is used with a value of 0.8 to enhance convergence.

For the trading strategy, the model operates in a monthly trading mode, where each trading period is defined as 21 days and the fee per trade is set at 0.001. The model handles up to 4 trades per round and is evaluated over 80,000 rounds. All computations are performed on a CUDA-enabled device.

doi:10.6342/NTU202403155

4.5 Roadmap of Experiments

This study aims to evaluate and enhance the DeepTrader model's performance through a series of experiments focusing on various aspects such as market sensitivity, model adaptation to different markets, and the integration of advanced architectures. The experiments are structured as follows:

4.5.1 Experiment 1: Analysis of Market and ρ Interaction

This study aims to conduct a comprehensive analysis of the DeepTrader model's performance in the U.S. market, focusing specifically on its sensitivity to the proportion of short-selling funds, denoted as ρ. To achieve this, the research will explore various aspects of the model's behavior and effectiveness under different market conditions. The following methodology will guide the investigation:

1. **Parameter Sensitivity Analysis**: The impact of different hyperparameters on the model's performance will be evaluated, with a particular emphasis on ρ, which represents the proportion of short-selling funds in the market.

2. **Market Condition Analysis**: The robustness and adaptability of the model will be assessed under varying market phases, including bull, bear, and sideways markets. This analysis will include periods of high volatility and stable market conditions.

3. **Reason for Adjustment**: Initial testing revealed that the model's ability to learn market dynamics was not sufficiently pronounced. By incorporating ρ as a variable, the study seeks to enhance the model's capability to capture and adapt to variations influenced by short-selling activities, which can significantly impact market

doi:10.6342/NTU202403155

behavior.

4. **Expected Outcomes**: The study anticipates that analyzing the interaction with ρ will provide insights into whether adjusting for short-selling dynamics can improve the model's performance metrics, such as predictive accuracy, risk management, and overall profitability.

4.5.2 Experiment 2: Test in the Taiwanese Market

The objective of this experiment is to assess the applicability and performance of the DeepTrader model within the Taiwanese market context, thereby exploring its adaptability across different market environments. Specifically, this study aims to determine how well the model, initially designed with Graph Convolutional Networks (GCN) and spatial attention mechanisms, performs when applied to Taiwanese market data. This experiment is further divided into two parts.

In Experiment 2-1, we first apply the original DeepTrader model directly to the Taiwanese market to observe its performance. The results of this initial application will provide insights into the model's effectiveness and any potential limitations when operating in this new market environment.

Following the observations from Experiment 2-1, Experiment 2-2 involves modifying the original model architecture to better suit the Taiwanese market. Based on the insights gained from the initial test, we make adjustments to enhance the model's performance and adaptability to the specific characteristics of the Taiwanese market. This iterative approach aims to refine the model and improve its predictive accuracy and robustness in a new market context.The methodology for this experiment includes the following

doi:10.6342/NTU202403155

steps:

1. **Model Modification**: The original DeepTrader model incorporates GCN to learn complex relationships within the data. However, preliminary tests using Taiwanese market data revealed that the weights associated with GCN tended to converge towards zero. This observation suggested that the GCN component might not effectively capture the relevant market dynamics specific to the Taiwanese context.

2. **Experimental Adjustment**: Based on these observations, an experimental decision was made to remove the GCN component from the DeepTrader model. This adjustment aims to investigate whether the removal of GCN would enhance the model's performance in the Taiwanese market setting.

3. **Performance Evaluation**: The performance of the modified DeepTrader model will be assessed using key financial metrics, including annual return, Sharpe ratio, and maximum drawdown. These metrics will provide insights into the effectiveness of the model without the GCN component in accurately predicting market behavior and generating investment strategies tailored to the Taiwanese market.

4. **Market-Specific Challenges**: Additionally, specific challenges inherent to the Taiwanese market, such as market size, liquidity constraints, and unique investor behaviors, will be identified and considered in the analysis. Understanding these factors is crucial for evaluating how well the modified DeepTrader model can adapt to and perform within this particular market environment.

doi:10.6342/NTU202403155

4.5.3 Experiment 3: Improvements Based on Transformer

This experiment aims to integrate a Transformer-based model architecture into the DeepTrader framework and evaluate its performance within the Taiwanese market context. Specifically, this involves replacing the Graph Convolutional Networks (GCN) and Long Short-Term Memory (LSTM) components of the original DeepTrader model with Transformers. This modern architecture is renowned for its ability to capture long-range dependencies and effectively model sequences. The methodology for this experiment includes the following steps:

1. **Model Architecture Adjustment**: The GCN and LSTM components of the original DeepTrader model will be substituted with Transformer layers. These layers use self-attention mechanisms to learn dependencies across both temporal and spatial dimensions in the market data.

2. **Model Training and Evaluation**: The Transformer-based model will be trained using historical Taiwanese market data. The focus will be on optimizing parameters and fine-tuning the architecture to improve predictive accuracy and stability.

3. **Performance Comparison**: A comparative analysis will be performed between the original DeepTrader model and the Transformer-based model. Key metrics, such as annual return, Sharpe ratio, and maximum drawdown, will be evaluated to assess improvements in model performance, particularly in terms of adaptability to dynamic market conditions and robustness against noise and volatility.

doi:10.6342/NTU202403155

4.5.4 Experiment 4: Model Retraining with Updated Information

This experiment investigates the benefits of periodically retraining the DeepTrader model with updated market information to assess its impact on predictive accuracy and model adaptability. The goal is to determine whether continuous updates to the model can enhance its performance by reflecting changing market dynamics. The methodology includes the following steps:

1. **Retraining Strategy**: The model is initially trained on a historical dataset covering ten years of market data. It is then tested on a separate validation set consisting of the following two years of data. After each testing phase, the model parameters are updated with the most recent data, effectively retraining the model to incorporate the latest market information.

2. **Performance Evaluation**: The periodically retrained model's performance is compared against a static model that remains unchanged after initial training. Metrics such as prediction accuracy, risk-adjusted returns, and consistency in performance are analyzed to evaluate how well retraining improves the model's adaptability and responsiveness to evolving market conditions.

3. **Impact Assessment**: By assessing how the model's performance evolves over time with continuous retraining, the study aims to gain insights into whether frequent updates enhance the model's ability to capture new trends, mitigate risks, and maintain robust performance across different market cycles.

By conducting these experiments, this study aims to enhance the robustness and adaptability of the DeepTrader model across different markets, particularly focusing on

doi:10.6342/NTU202403155

the Taiwanese market's unique challenges. The introduction of the Transformer architecture and the strategy of periodic retraining are expected to address the limitations of the original model and improve its performance in dynamic and complex financial environments.

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

Chapter 5 Experiments and Results

This chapter offers a detailed analysis of the experiments conducted as outlined in Chapter 4. It begins by thoroughly examining the process of each experiment, including the training, validation, and testing phases. The chapter provides a clear depiction of how these models were developed and evaluated, emphasizing the performance metrics used to assess their effectiveness, such as accuracy, return rates, and risk measures.

Following this, the chapter delves into comparative analyses, contrasting the performance of the newly developed models with existing ones. This includes a discussion on the enhancements achieved by integrating Transformer-based architectures and other improvements. The chapter then explores the implications of these findings, highlighting their impact on financial modeling strategies and how they contribute to the development of more effective trading strategies and investment decisions.

The experiments discussed cover a range of topics, from analyzing market interactions to testing the benefits of Transformer-based models and assessing the impact of periodic retraining with updated information. By integrating these aspects, the chapter provides a comprehensive understanding of the experimental process and its contributions to advancing financial market analysis and trading strategies.

doi:10.6342/NTU202403155

5.1 Experiment 1: Analysis of Market and ρ Interaction

This experiment aimed to analyze how variations in ρ, representing the proportion of short-selling funds, influence the DeepTrader (DT) model's performance in the U.S. market. The findings revealed that when ρ was set to 1 (indicating 100% short positions), the model exhibited a higher winning rate. This suggests that the model may benefit from market conditions where short-selling is prevalent, possibly due to better exploitation of market downturns or corrections.

Conversely, when ρ was adjusted to balance long and short positions, the model often displayed lower overall risk. This observation aligns with conventional portfolio management strategies that aim to diversify risk exposure. Manual adjustments to ρ underscored the limitations of the original Market Scoring Unit (MSU) in effectively learning and adapting to varying ρ levels, indicating a need for model improvements in capturing nuanced market dynamics.

Furthermore, comparative analyses against market benchmarks over different time frames highlighted DeepTrader's inconsistency in outperforming the market consistently. While the model occasionally demonstrated strong performance in specific periods, its overall predictability and reliability for investors were compromised. This variability underscores the importance of enhancing the model's winning rate to increase its practical utility in investment decision-making.

doi:10.6342/NTU202403155

5.1.1 Training, Validation and Test Intervals

Table 5.1 lists the intervals used for training, validation and test during the experiment:

Table 5.1: Training, Validation, and Test Intervals of Exp1

Interval Type	Start Date	End Date
Training	2000/01/01	2007/10/31
Validation	2007/11/01	2015/11/30
Test	2015/12/01	2023/12/31

5.1.2 Performance Metrics and Observations

The performance of strategies with different ρ settings is presented in Figures 5.1 and 5.2. From Figure 5.1, it is evident that the strategies with ρ manually set to 1 consistently performed the best, with the highest winning rates. Figure 5.2 and Table 5.2 further illustrate that setting ρ to 1 achieved the highest annual return rates for DT (DeepTrader) and DTWM (DeepTrader without MSU). However, the maximum drawdown (MDD) was typically observed when ρ was set to 0.5, indicating a balanced approach between long and short positions.

Table 5.2: The metric values corresponding to each strategy in US stock market of Exp1

	ASR (↑)	**ARR (↑)**	**MDD (↓)**
DT	0.65	10.39%	32.41%
DT (ρ = 0)	0.62	10.11%	36.35%
DT (ρ = 0.5)	0.82	13.32%	25.27%
DT (ρ = 1)	0.82	15.65%	30.81%
DTWM	0.62	9.93%	19.84%
DTWM (ρ = 0)	0.32	4.41%	50.91%
DTWM (ρ = 1)	0.79	15.04%	29.73%
DJIA	0.46	6.36%	42.01%

Figure 5.3 and Table 5.3 demonstrate that the winning rate was generally highest when ρ was set to 1. However, the original DeepTrader strategy showed low winning rates

doi:10.6342/NTU202403155

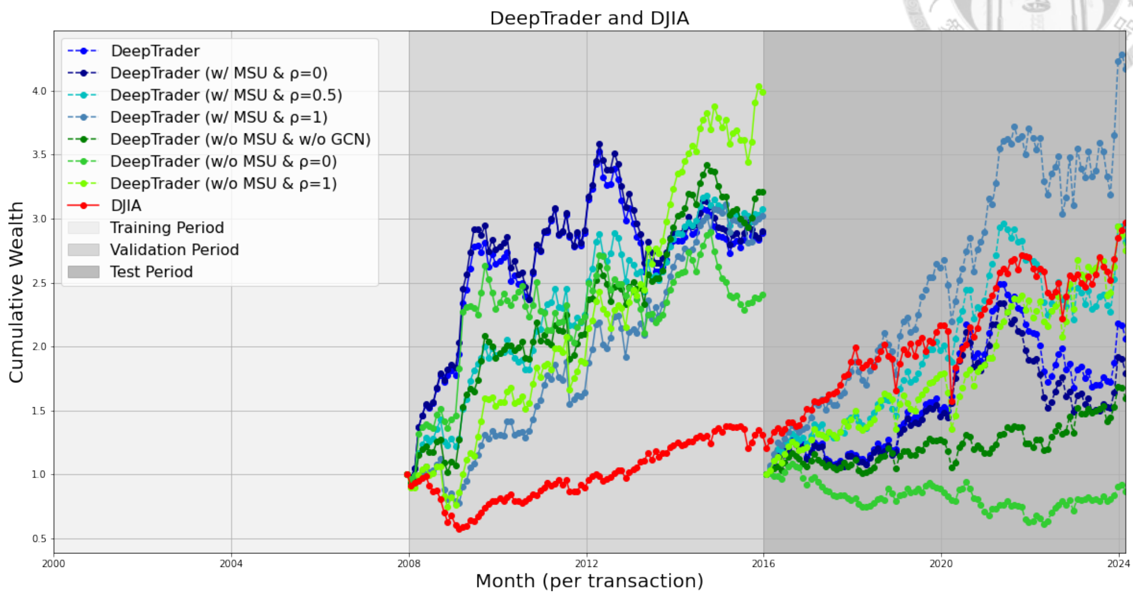

Figure 5.1: Performance with different ρ settings in US market

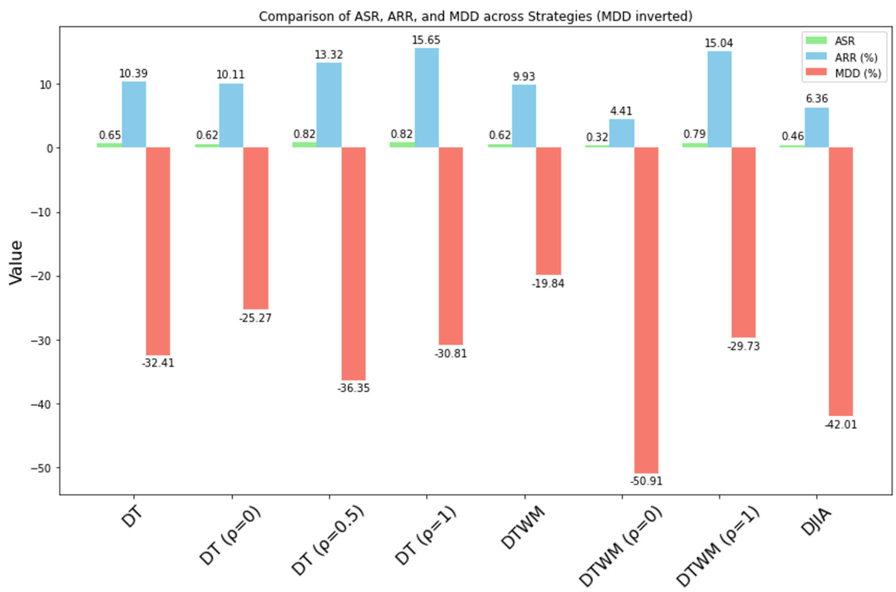

Figure 5.2: Metric values comparison of Exp1

doi:10.6342/NTU202403155

regardless of the timeframe, whether considering each trade individually or extending the period to one year.

Table 5.3: The winning rate of different strategies compared to the market at different frequencies of Exp1

	Per trade	3 month	6 month	12 month
DT	50.52%	43.75%	29.41%	37.50%
DT ($\rho = 1$)	58.52%	53.12%	64.71%	75.00%
DTWM	48.45%	40.62%	47.06%	37.50%
DTWM ($\rho = 1$)	57.73%	46.88%	47.06%	62.50%

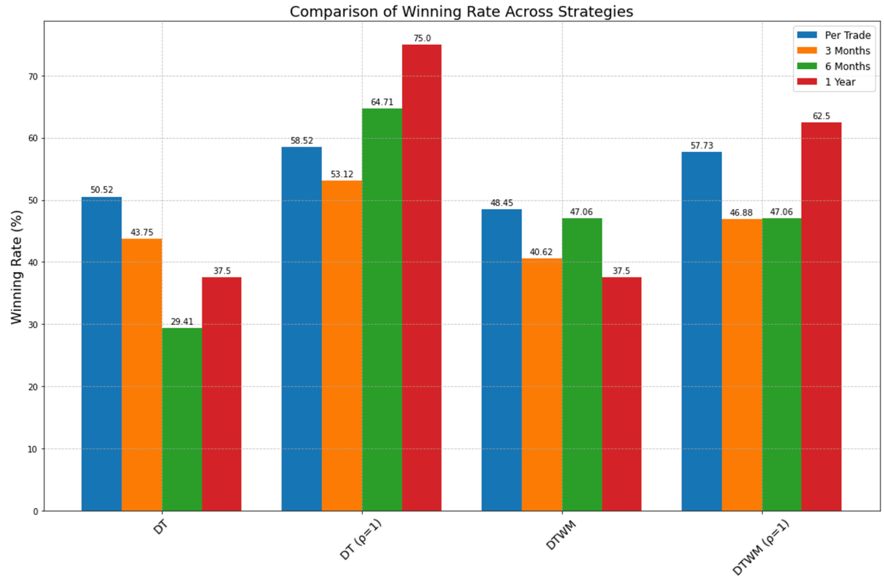

Figure 5.3: Winning rate comparison of Exp1

5.1.3 Comparative Analysis

Figures 5.4, 5.5, Table 5.4 and Table 5.5 provide a comparative analysis of each strategy's annual return rates. They reflect that the original DeepTrader strategy, characterized by high volatility, achieved higher returns in specific trades but lacked consistency. This high-risk, high-reward nature of the strategy is not ideal for conservative investors, who require more stable and predictable returns over time. They are unlikely to prefer a strategy where the timing of significant profits is uncertain.

doi:10.6342/NTU202403155

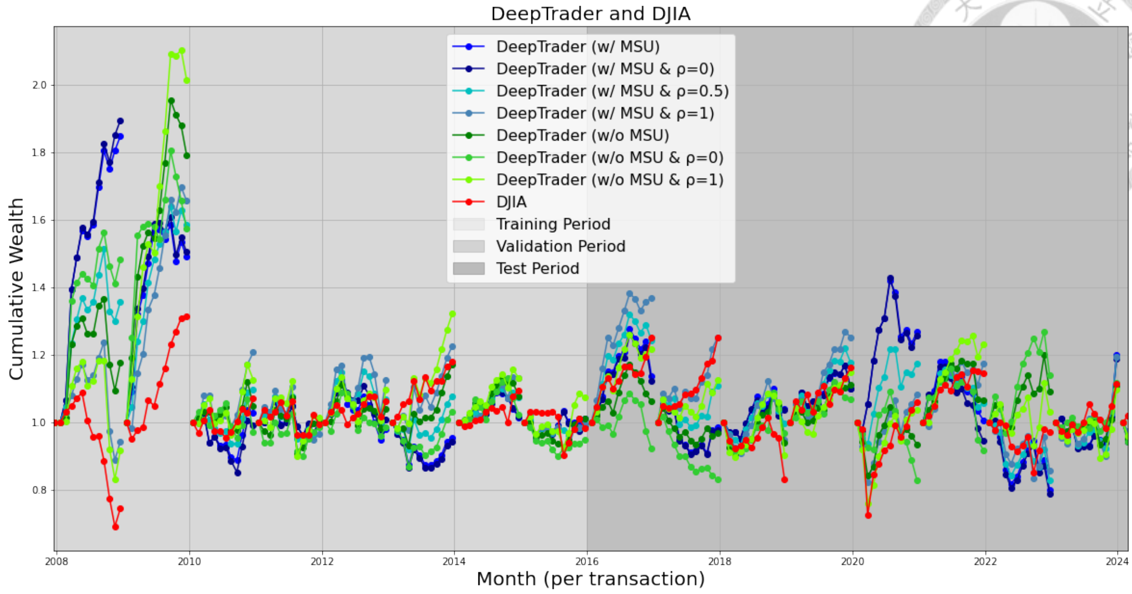

Figure 5.4: Cumulative wealth every year corresponding to each strategy in US stock market

Table 5.4: Annual return of Exp1 from 2008 to 2015

	2008	2009	2010	2011	2012	2013	2014	2015
DT	69.52%	37.73%	7.28%	8.75%	-8.46%	-4.49%	1.12%	-8.53%
DTWM	12.77%	53.66%	5.74%	16.48%	5.38%	17.15%	4.38%	-6.44%
DJIA	-30.62%	30.34%	15.00%	2.74%	3.73%	17.91%	13.53%	1.52%

Table 5.5: Annual return of Exp1 from 2016 to 2023

	2016	2017	2018	2019	2020	2021	2022	2023
DT	15.59%	-7.61%	21.96%	9.24%	40.83%	-7.59%	-13.24%	1.45%
DTWM	8.89%	-4.03%	11.36%	6.22%	-1.42%	0.46%	19.08%	3.72%
DJIA	20.02%	24.44%	-10.79%	14.16%	8.32%	21.20%	-5.65%	-3.03%

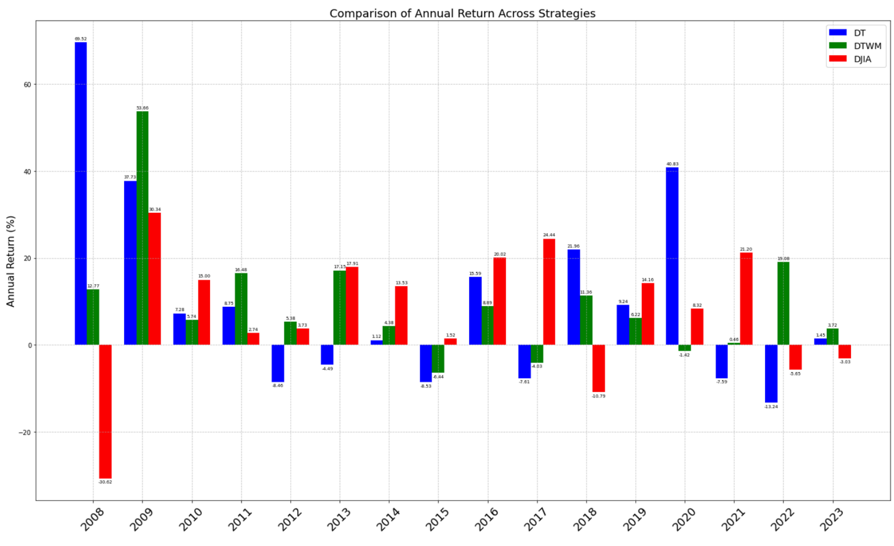

Figure 5.5: Annual return comparison of Exp1

doi:10.6342/NTU202403155

5.1.4 Discussion and Implications

The detailed analysis and results from the experiment provide several key insights into the DeepTrader model's performance:

1. **Performance with High ρ Values**: When ρ was set to 1, indicating 100% short positions, DeepTrader showed a higher winning rate and annual returns, suggesting that the model excels in short-selling strategies, particularly during market downturns.

2. **Risk Management with Balanced ρ Values**: Setting ρ to 0.5, balancing long and short positions, resulted in the lowest drawdowns, indicating better risk management and aligning with traditional portfolio diversification strategies.

3. **Model Inconsistencies**: The original DeepTrader model exhibited significant variability in performance, particularly when evaluated over longer periods. This inconsistency reduces its practical utility for investors seeking reliable and predictable returns.

4. **Need for Model Improvements**: The limitations observed in the original Market Scoring Unit (MSU) in adapting to different ρ levels highlight the necessity for model improvements to better capture market dynamics and enhance overall performance.

The findings underscore the potential of integrating advanced techniques such as Transformer networks to improve DeepTrader's adaptability and performance across various market conditions. Further research and development in these areas could significantly enhance the model's utility and robustness for practical investment applications.

doi:10.6342/NTU202403155

5.2 Experiment 2: Test in the Taiwanese Market

The direct application of the original DeepTrader (DT) model to the Taiwanese market yielded suboptimal results. This outcome showed in Figures 5.6, 5.7 and Table 5.7 suggests that the model's effectiveness, originally validated in the U.S. market, does not seamlessly transfer to different market environments due to unique market characteristics such as liquidity, investor behavior, and regulatory influences. The following section presents the results of Experiment 2-1, where we first apply the original DeepTrader model directly to the Taiwanese market to observe its performance. The results of this initial application provide insights into the model's effectiveness and any potential limitations when operating in this new market environment.

Table 5.6 lists the intervals used for training, validation, and test during the experiment:

Table 5.6: Training, Validation, and Test Intervals of Exp2-1

Interval Type	Start Date	End Date
Training	2000/01/01	2008/01/31
Validation	2008/02/01	2015/10/31
Test	2015/11/01	2023/12/31

Table 5.7: The metric values corresponding to each strategy in TW stock market of Exp2-1

	ASR (↑)	**ARR (↑)**	**MDD (↓)**
DT	0.47	6.97%	42.50%
0050.TW	0.38	5.79%	52.69%

Upon removing the Graph Convolutional Network (GCN) component from the DeepTrader architecture, significant improvements in learning efficacy were observed in the Taiwanese market context. The decision to remove GCN was based on empirical evidence showing that GCN weights tended to converge towards zero when applied to Taiwanese

doi:10.6342/NTU202403155

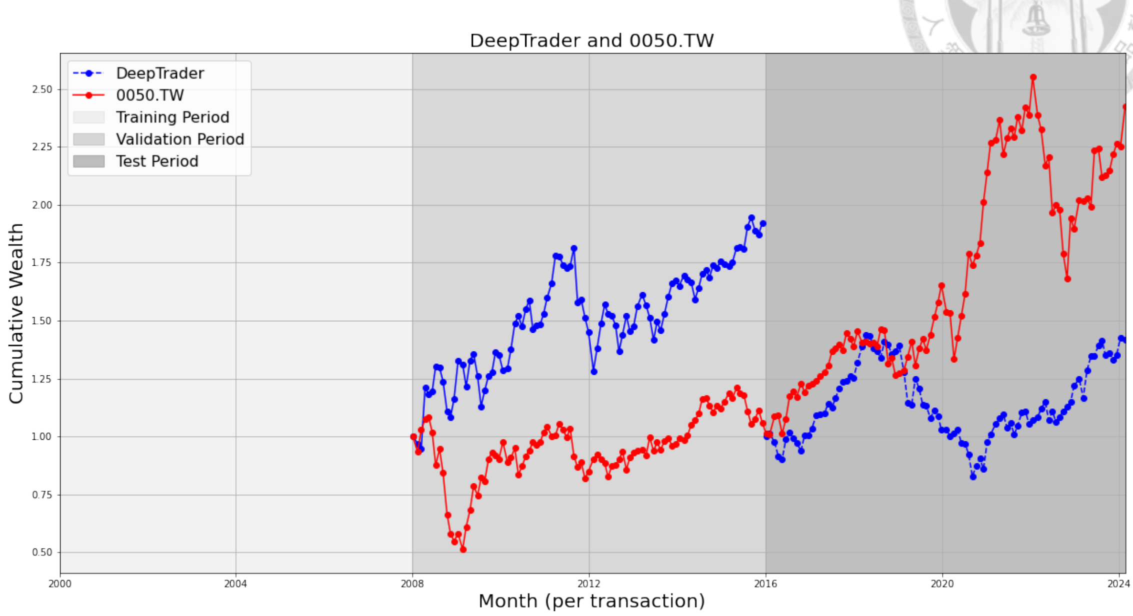

Figure 5.6: DeepTrader in TW market

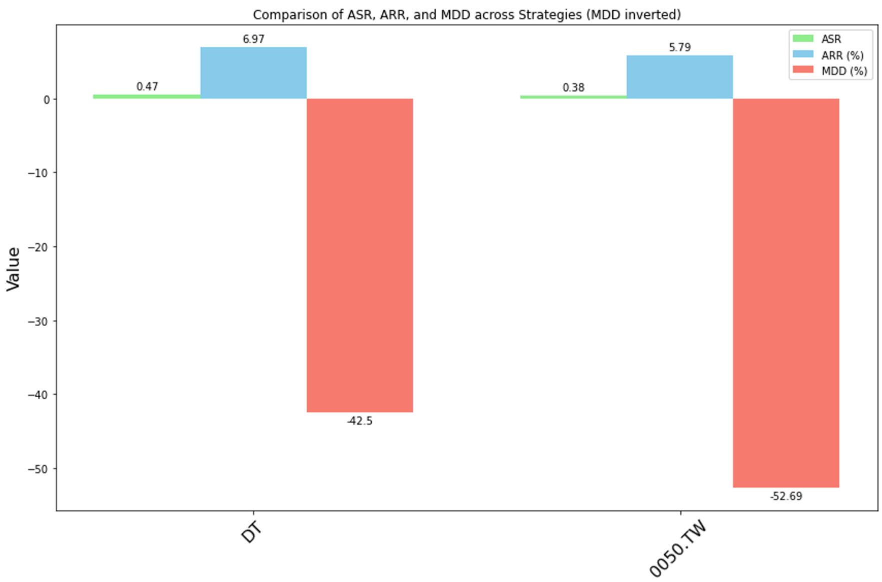

Figure 5.7: Metric values comparison of Exp2-1

doi:10.6342/NTU202403155

market data. This suggests that the GCN component was not effectively capturing the relational data in this specific market. There are several potential reasons for this:

1. **Market Size and Liquidity**: The Taiwanese market is smaller and less liquid compared to the U.S. market. This reduced liquidity can lead to less reliable relational data, making it difficult for the GCN to learn meaningful patterns.

2. **Investor Behavior**: Taiwanese investors often exhibit different trading behaviors, such as higher levels of retail participation and different risk appetites compared to U.S. investors. These behaviors can result in different market dynamics that are not as effectively captured by the GCN.

3. **Regulatory Influences**: Regulatory frameworks and market mechanisms in Taiwan differ from those in the U.S., which can impact how information is disseminated and how stocks are correlated. These differences might render the relational data less predictive, thus reducing the efficacy of the GCN.

In contrast, the U.S. market's larger size, higher liquidity, and more stable investor behavior patterns provide a richer and more reliable relational data set, allowing the GCN to effectively capture long-term dependencies and relationships between stocks.

Based on the results of Experiment 2-1, we decided to remove the GCN component to see if this adjustment would enhance the model's ability to adapt to and capture relevant market dynamics more effectively in the Taiwanese market. This adjustment is detailed in Experiment 2-2. By removing the GCN component, we aimed to improve the overall performance metrics of the model in the Taiwanese market.

doi:10.6342/NTU202403155

5.2.1 Training, Validation and Test Intervals

Table 5.8 lists the intervals used for training, validation, and test during the experiment:

Table 5.8: Training, Validation, and Test Intervals of Exp2-2

Interval Type	Start Date	End Date
Training	2000/01/01	2007/12/31
Validation	2008/01/01	2015/12/31
Test	2016/01/01	2024/02/28

Figure 5.8 illustrates the performance comparison of different strategies in the Taiwanese market after removing the GCN component. It is evident that the strategies without MSU and manually setting ρ to 1 achieved the best performance:

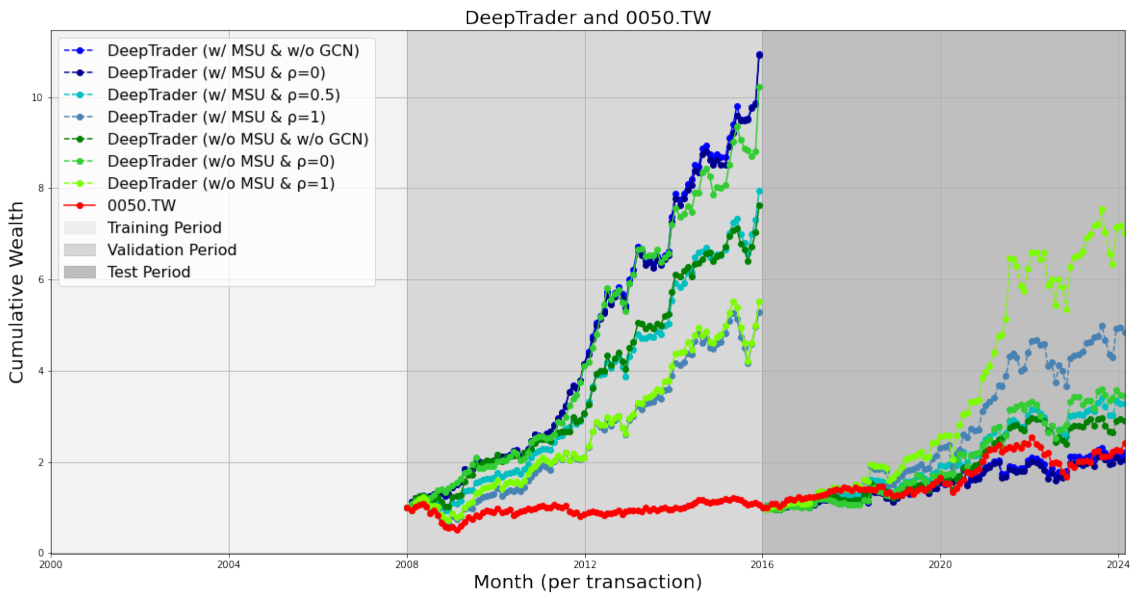

Figure 5.8: The performance of DeepTrader (without GCN) with different ρ settings in TW Market

5.2.2 Performance Metrics and Observations

Table 5.9 and Figures 5.9 present the performance metrics and observations for DT (DeepTrader) and DTWM (DeepTrader without MSU).

doi:10.6342/NTU202403155

Table 5.9: The metric values corresponding to each strategy under DeepTrader without GCN in TW stock market of Exp2-2

	ASR (↑)	ARR (↑)	MDD (↓)
DT	1.22	22.21%	19.22%
DT ($\rho = 0$)	1.19	21.67%	21.26%
DT ($\rho = 0.5$)	1.21	22.72%	25.66%
DT ($\rho = 1$)	0.99	22.49%	48.27%
DTWM	1.13	21.47%	20.19%
DTWM ($\rho = 0$)	1.27	25.17%	24.37%
DTWM ($\rho = 1$)	1.07	25.76%	40.51%
0050.TW	0.38	5.58%	52.02%

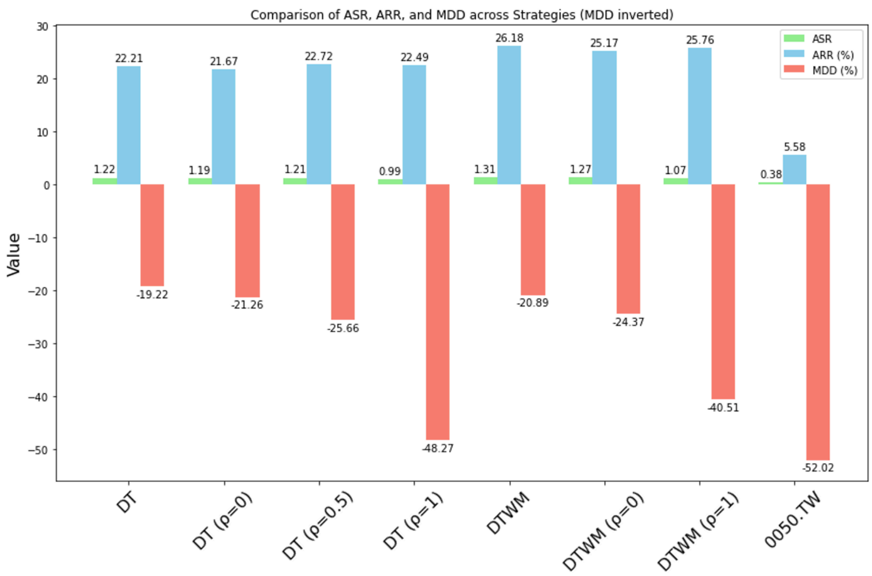

Figure 5.9: Metric values comparison of Exp2-2

doi:10.6342/NTU202403155

5.2.3 Comparative Analysis

Figures 5.12 and 5.10 show that setting ρ to 1 achieved the highest annual return and winning rate in the Taiwanese market. However, this setting also resulted in a higher MDD compared to $\rho = 0.5$, indicating increased risk for higher returns:

Table 5.10: The winning rate of different strategies compared to the market at different frequencies of Exp2-2

	Per trade	**3 month**	**6 month**	**12 month**
DT	55.85%	43.75%	35.29%	37.50%
DT ($\rho = 1$)	62.23%	53.12%	58.82%	75.00%
DTWM	58.51%	53.12%	41.18%	62.50%
DTWM ($\rho = 1$)	60.64%	62.50%	52.94%	75.00%

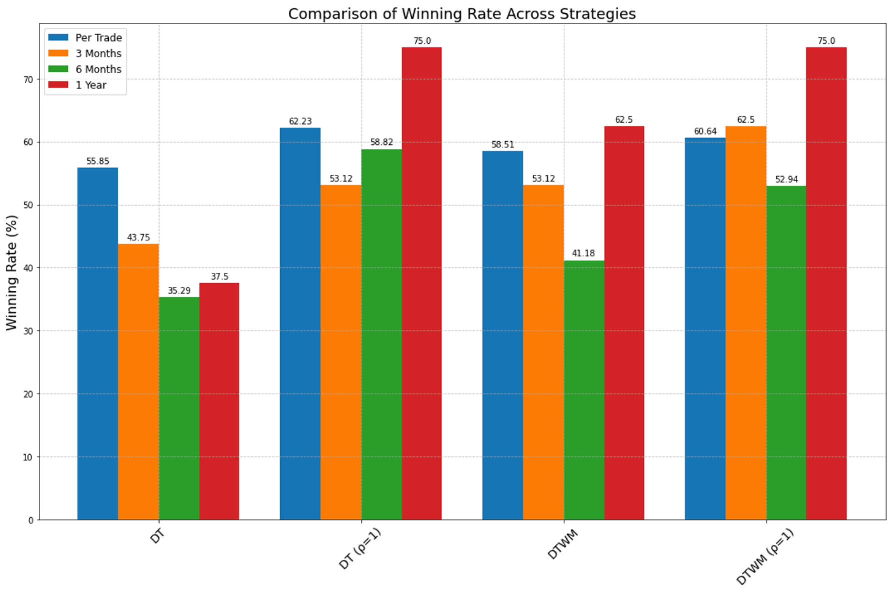

Figure 5.10: Winning rate comparison of Exp2-2

Table 5.11: Annual return of Exp2-2 from 2008 to 2015

	2008	2009	2010	2011	2012	2013	2014	2015
DT	29.48%	40.15%	22.33%	62.36%	22.87%	22.61%	11.01%	25.89%
DTWM	3.66%	48.94%	26.89%	27.57%	19.51%	30.02%	7.40%	21.50%
0050.TW	-47.13%	78.22%	4.26%	-17.91%	1.13%	4.60%	14.19%	-7.36%

doi:10.6342/NTU202403155

Table 5.12: Annual return of Exp2-2 from 2016 to 2023

	2016	2017	2018	2019	2020	2021	2022	2023
DT	3.58%	8.98%	3.05%	28.81%	18.31%	7.94%	1.16%	6.31%
DTWM	6.37%	16.19%	16.06%	35.96%	43.97%	45.71%	1.12%	7.14%
0050.TW	22.59%	12.36%	-12.22%	31.98%	32.96%	14.00%	-26.91%	10.31%

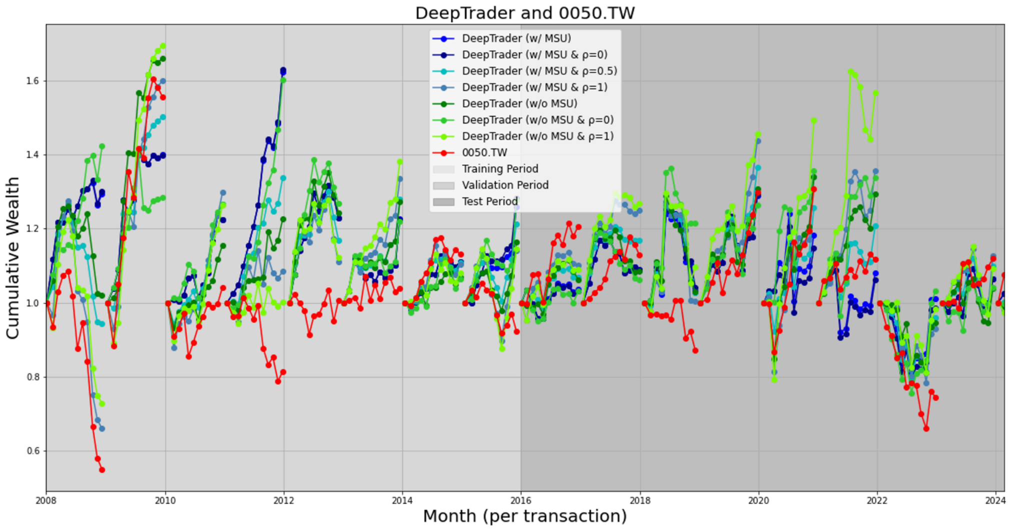

Figure 5.11: Cumulative wealth every year corresponding to each strategy in TW stock market of Exp2-2

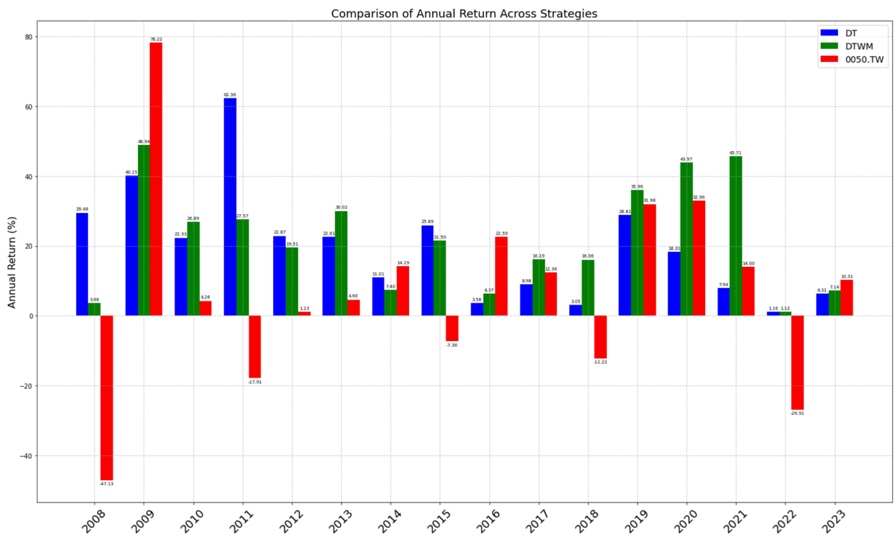

Figure 5.12: Annual return comparison of Exp2-2

doi:10.6342/NTU202403155

5.2.4 Discussion and Implications

The experiment highlights several critical findings regarding the adaptation of the DeepTrader model to the Taiwanese market:

1. **Market-Specific Challenges**: The Taiwanese market's unique characteristics such as liquidity constraints and investor behavior posed challenges for the original DeepTrader model's effectiveness.

2. **Impact of GCN Removal**: Removing the GCN component led to significant improvements in learning efficacy and overall model performance in capturing Taiwanese market dynamics.

3. **Optimal Strategies**: Strategies without MSU and manually setting ρ to 1 consistently outperformed other configurations in terms of annual return and winning rate, despite higher associated risks.

4. **Future Directions**: Further research could focus on refining model architectures tailored to specific market conditions and integrating alternative learning algorithms to enhance adaptability and performance.

These conclusions underscore the importance of adapting deep learning models like DeepTrader to local market conditions and refining them iteratively to achieve robust performance across diverse financial environments.

doi:10.6342/NTU202403155

5.3 Experiment 3: Improvements Based on Transformer

The experiment involved replacing the Graph Convolutional Network (GCN) and Long Short-Term Memory (LSTM) components of the original DeepTrader model with Transformer architecture, renowned for its robust sequence modeling capabilities. Transformers excel in capturing long-range dependencies in data through self-attention mechanisms, which are crucial in financial markets where historical context significantly influences future trends.

Transformers utilize self-attention mechanisms to weigh the significance of different parts of the input data, allowing the model to focus on the most relevant information for prediction. This characteristic is particularly beneficial in financial markets, where relationships between different stocks and market indicators can be complex and non-linear. Unlike GCN and LSTM, Transformers do not require predefined structures to handle these relationships, providing a significant advantage.

5.3.1 Training, Validation, and Test Intervals

Table 5.13 lists the intervals used for training, validation, and test during the experiment:

Table 5.13: Training, Validation, and Test Intervals of Exp3

Interval Type	Start Date	End Date
Training	2000/01/01	2007/12/31
Validation	2008/01/01	2015/12/31
Test	2016/01/01	2024/02/28

doi:10.6342/NTU202403155

5.3.2 Performance Metrics and Observations

Figure 5.13, 5.14 and Table 5.14 illustrate the performance comparison of different strategies after incorporating the Transformer architecture. It is evident that replacing the GCN in the Asset Scoring Unit (ASU) with Transformers (DTTF (ASU)) significantly improved performance. Further enhancements were observed when the LSTM in the Market Scoring Unit (MSU) was also replaced by Transformers (DTTF (ASU & MSU)), compared to DT (DeepTrader) and DTWM (DeepTrader without MSU).

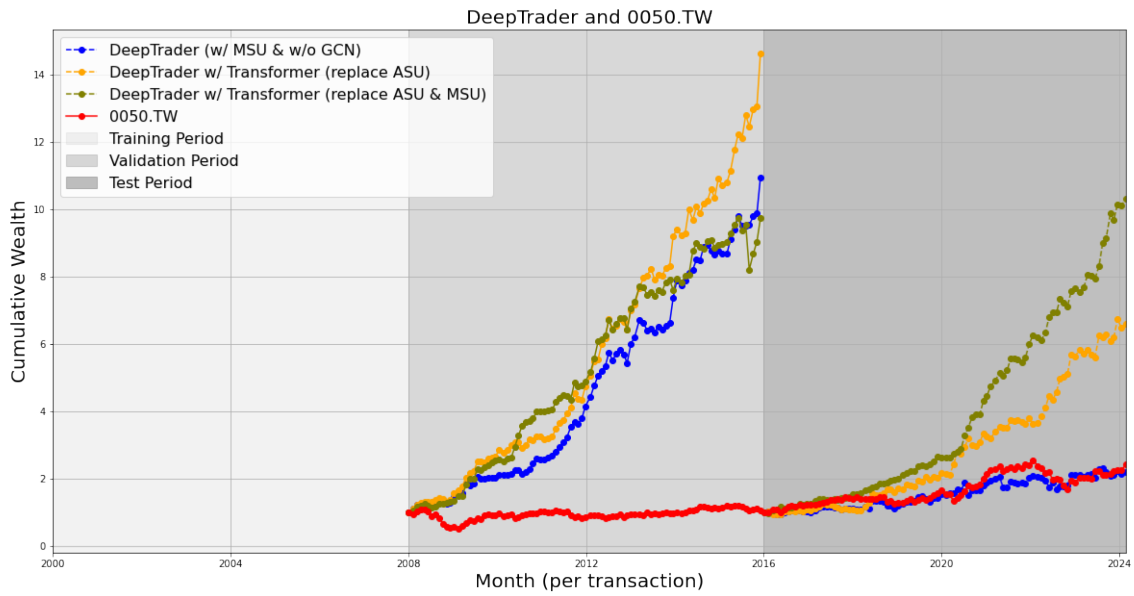

Figure 5.13: Performance with Transformer replacements in TW market

Table 5.14: The metric values corresponding to each strategy under DeepTrader with Transformer replacements in TW stock market of Exp3

	ASR (↑)	ARR (↑)	MDD (↓)
DT	1.22	22.21%	19.22%
DTWM	1.13	21.47%	20.19%
DTTF (ASU)	1.34	24.66%	21.83%
DTTF (ASU & MSU)	1.69	25.83%	15.79%
0050.TW	0.38	5.79%	52.69%

doi:10.6342/NTU202403155

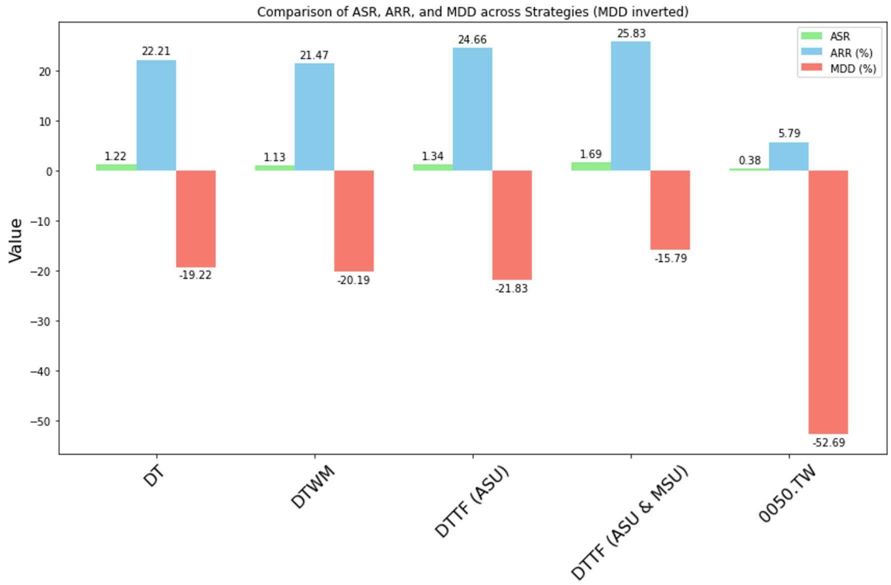

Figure 5.14: Metric values comparison of Exp3

5.3.3 Comparative Analysis

Figures 5.15 and 5.17 show the winning rate and annual return for different strategies. It can be seen that using Transformers in both ASU and MSU not only achieved the highest annual return but also significantly reduced the maximum drawdown, indicating a more robust and reliable performance:

Table 5.15: The winning rate of different strategies compared to the market at different frequencies of Exp3

	Per trade	3 month	6 month	12 month
DT	55.85%	43.75%	35.29%	37.50%
DTWM	58.51%	53.12%	41.18%	62.50%
DTTF (ASU)	57.98%	59.38%	41.18%	37.50%
DTTF (ASU & MSU)	57.98%	57.14%	61.11%	75.00%

Table 5.16, Table 5.17 and Figures 5.16 to 5.17 present the yearly return comparison. It is evident that the Transformer-based architecture consistently achieved the highest annual returns across different years:

doi:10.6342/NTU202403155

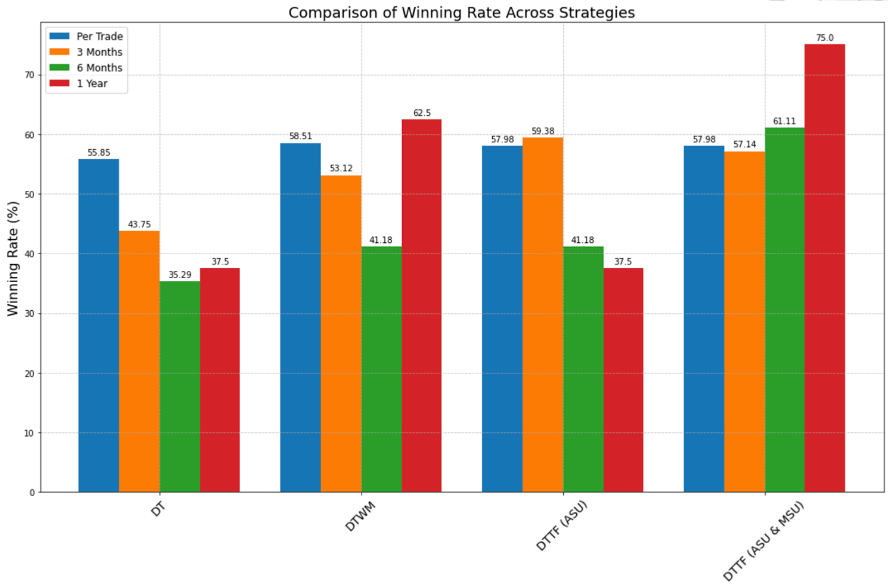

Figure 5.15: Winning rate comparison of Exp3

Table 5.16: Annaul return of Exp3 from 2008 to 2015

	2008	2009	2010	2011	2012	2013	2014	2015
DT	29.48%	40.15%	22.33%	62.36%	22.87%	22.61%	11.01%	25.89%
DTWM	1.97%	66.10%	15.56%	22.64%	24.37%	27.32%	6.77%	16.31%
DTTF (ASU)	36.81%	69.68%	13.80%	49.48%	27.06%	31.34%	16.27%	36.39%
DTTF (ASU & MSU)	32.68%	86.78%	54.96%	22.19%	24.75%	7.81%	12.81%	8.59%
0050.TW	-47.13%	78.22%	4.26%	-17.91%	1.13%	4.60%	14.19%	-7.36%

Table 5.17: Annual return of Exp3 from 2016 to 2023

	2016	2017	2018	2019	2020	2021	2022	2023
DT	3.58%	8.98%	3.05%	28.81%	18.31%	7.94%	1.16%	6.31%
DTWM	6.19%	8.50%	2.86%	30.79%	34.00%	29.43%	-5.52%	3.93%
DTTF (ASU)	7.85%	3.26%	-0.23%	22.71%	15.32%	18.65%	-1.74%	14.68%
DTTF (ASU & MSU)	18.67%	15.22%	9.09%	18.31%	41.54%	20.09%	3.76%	13.19%
0050.TW	22.59%	12.36%	-12.22%	31.98%	32.96%	14.00%	-26.91%	10.31%

doi:10.6342/NTU202403155

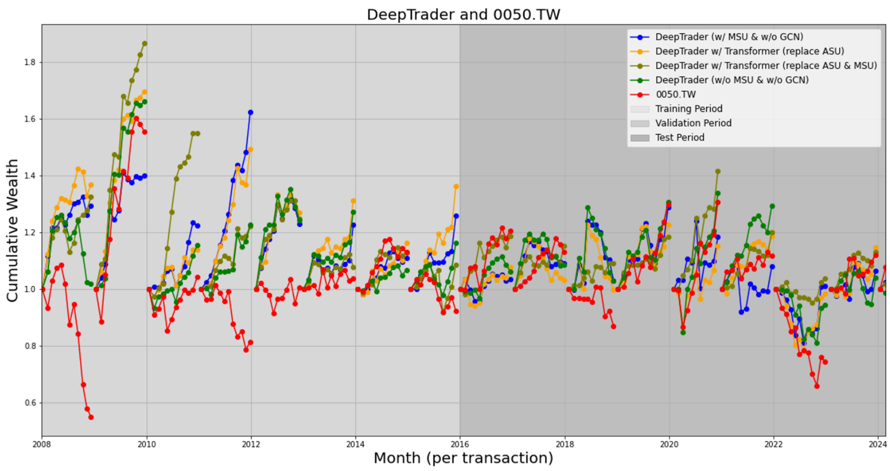

Figure 5.16: Cumulative wealth every year corresponding to each strategy in TW stock market of Exp3

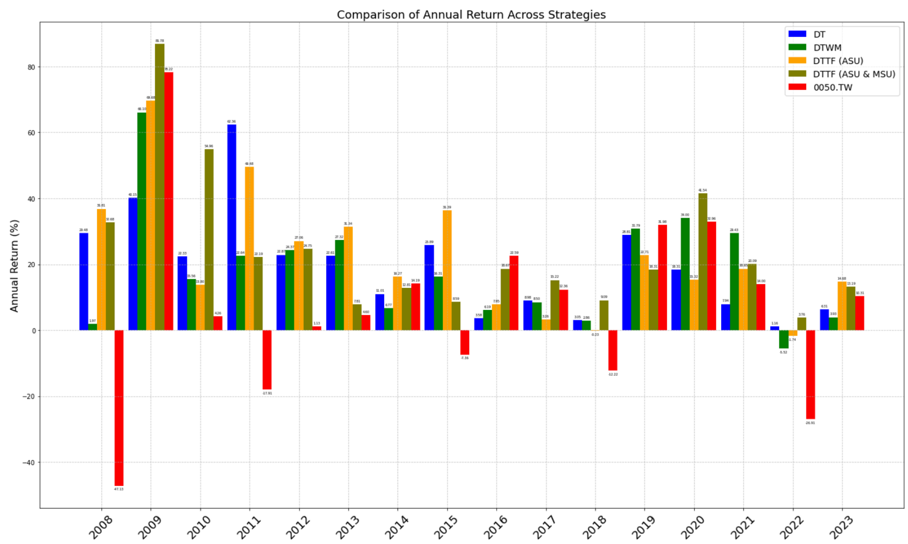

Figure 5.17: Annual return comparison of Exp3

doi:10.6342/NTU202403155

5.3.4 Discussion and Implications

The experiment highlights several critical findings regarding the adoption of Transformer architecture in the DeepTrader model:

1. **Market-Specific Adaptability**: The superior sequence modeling capabilities of Transformers significantly improved the model's performance in capturing complex market dynamics.

2. **Performance Enhancements**: Replacing both GCN in ASU and LSTM in MSU with Transformers resulted in higher annual returns and reduced maximum drawdown, indicating a more robust model.

3. **Optimal Strategies**: The Transformer-based architecture consistently outperformed the original model in terms of both winning rate and annual returns across different years, demonstrating its effectiveness in financial forecasting and investment strategies.

4. **Future Directions**: Further research could explore integrating more advanced reinforcement learning algorithms and incorporating market sentiment indices as features to enhance model performance further.

These conclusions underscore the importance of leveraging advanced architectures like Transformers to enhance the adaptability and robustness of financial models, ensuring they can effectively capture and respond to complex market conditions.

doi:10.6342/NTU202403155

5.4 Experiment 4: Model Retraining with Updated Information

The experiment focused on evaluating the benefits of periodically retraining the DeepTrader model with updated market information. Traditional models, once trained, remain static and often fail to adapt to evolving market conditions, leading to diminished performance over time. This experiment implemented a dynamic retraining strategy, updating the model's parameters monthly with the latest market data.

Initial findings indicated that the original DeepTrader model, despite its comprehensive training on a long historical dataset, struggled to maintain relevance as market dynamics shifted. Market conditions are inherently volatile, and static models may overlook recent trends or anomalies, leading to suboptimal performance.

By incorporating monthly updates to the model's weights based on the most recent data, significant performance improvements were observed. These enhancements included:

- **Increased Predictive Accuracy**: The model became more responsive to current market trends, improving its ability to make accurate predictions.
- **Reduced Risk Exposure**: Continuous updates helped in minimizing drawdowns, as the model could better anticipate and react to market downturns.
- **Improved Consistency**: The model consistently generated profitable investment decisions across different market cycles, demonstrating robustness and reliability.

The results highlighted the importance of adaptive learning and continuous model re-

doi:10.6342/NTU202403155

finement. In practical investment scenarios, the ability to adapt to new information ensures that the model remains relevant and accurate, thus enhancing its practical applicability.

5.4.1 Training, Validation, and Test Intervals of Exp4

The Figure 5.18 illustrates the shifting windows strategy for splitting the TW dataset into training, validation, and test periods. The detailed description is as follows:

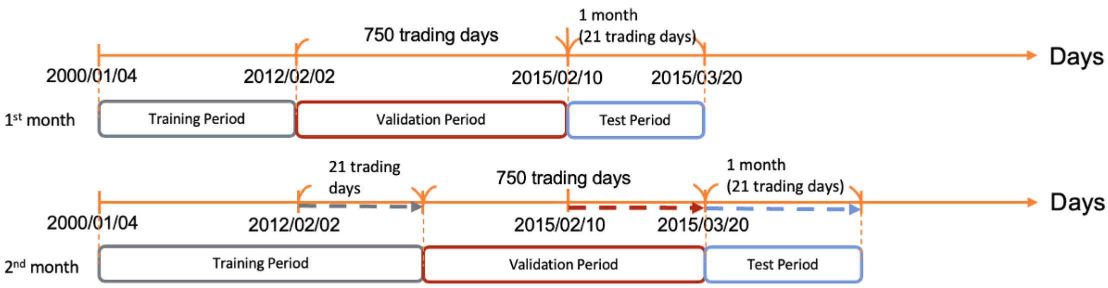

Figure 5.18: Training/Validation/Test splitting of Exp4 for TW dataset

1. **General Structure**: The timeline starts on January 4, 2000, and extends beyond March 20, 2015. The dataset is divided into two primary configurations, each representing a different month, with similar periods of training, validation, and test.

2. **1st Month Configuration**:

   - **Training Period**: Starts from January 4, 2000, and ends on February 2, 2012. This period encompasses 750 trading days, which is equivalent to approximately three years.

   - **Validation Period**: Immediately follows the training period, beginning on February 3, 2012, and ending on February 9, 2015. This period also spans 750 trading days.

   - **Test Period**: Starts right after the validation period, running from February 10, 2015, to March 20, 2015. This period consists of 21 trading days, representing

doi:10.6342/NTU202403155

roughly one month.

3. **2nd Month Configuration**:

    - **Training Period**: Identical to the 1st month configuration, starting from January 4, 2000, to February 2, 2012.
    - **Validation Period**: Begins on February 3, 2012, and concludes on February 9, 2015, covering the same 750 trading days as in the 1st month.
    - **Test Period**: Slightly shifts forward compared to the 1st month, commencing on February 10, 2015, and ending on March 20, 2015, lasting for 21 trading days.

4. **Moving Window Approach**: The key difference between the two configurations lies in the slight shift in the test period. Each month, the training and validation periods remain consistent, while the test period moves forward by one month (21 trading days). This approach ensures that the model is tested on different segments of the data while maintaining the training and validation phases constant.

5. **Purpose**: The moving window approach allows for a comprehensive evaluation of the model's performance over time, ensuring robustness and stability. By shifting the testing window, it captures different market conditions, providing a more thorough assessment of the model's predictive capabilities.

This detailed splitting method ensures that the model is both trained and validated on a substantial dataset, while the test period is small enough to provide frequent updates on the model's performance.

doi:10.6342/NTU202403155

5.4.2 Performance Metrics and Observations

Figure 5.19 to 5.20 and Table 5.18 illustrate the performance metrics after incorporating monthly weight updates with the Transformer architecture. It is evident that this approach, denoted as DT* (DeepTrader with monthly weight updates) and DTTF* (DeepTrader with Transformer and monthly weight updates), significantly improved returns while reducing risk (maximum drawdown) compared to DT (DeepTrader) and DTTF (DeepTrader with Transformer).

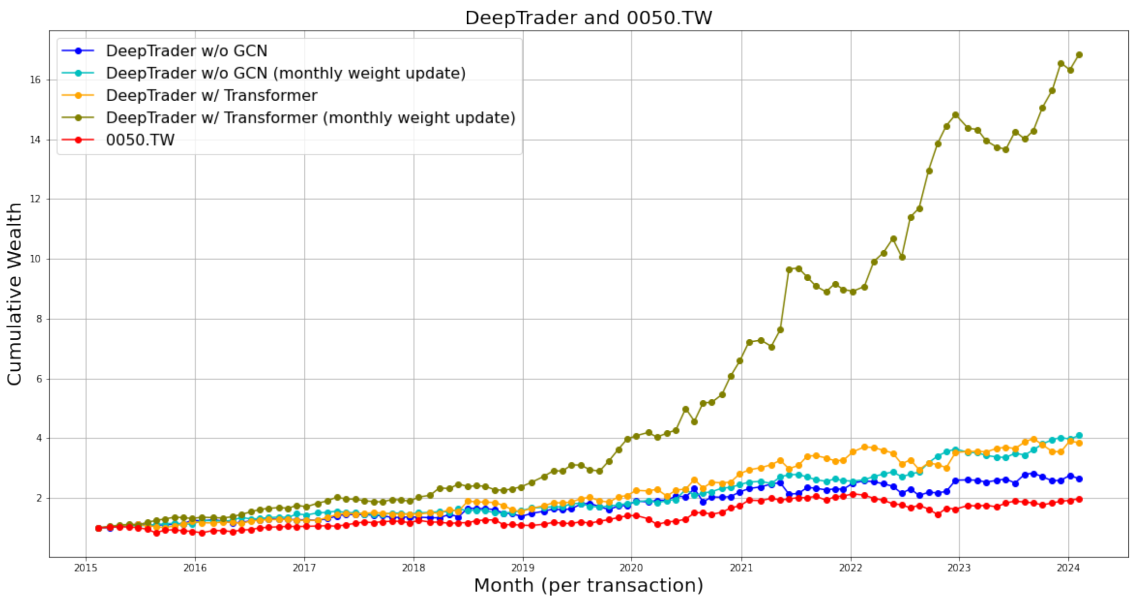

Figure 5.19: Performance with Transformer replacements and monthly updates in TW market

Table 5.18: The metric values corresponding to each strategy under DeepTrader with Transformer replacements and monthly updates in TW stock market of Exp4

	ASR (↑)	**ARR (↑)**	**MDD (↓)**
DT	0.66	11.74%	19.22%
DT*	1.30	17.78%	15.38%
DTTF	0.90	16.63%	20.50%
DTTF*	1.73	35.46%	9.93%
0050.TW	0.52	8.06%	32.40%

doi:10.6342/NTU202403155

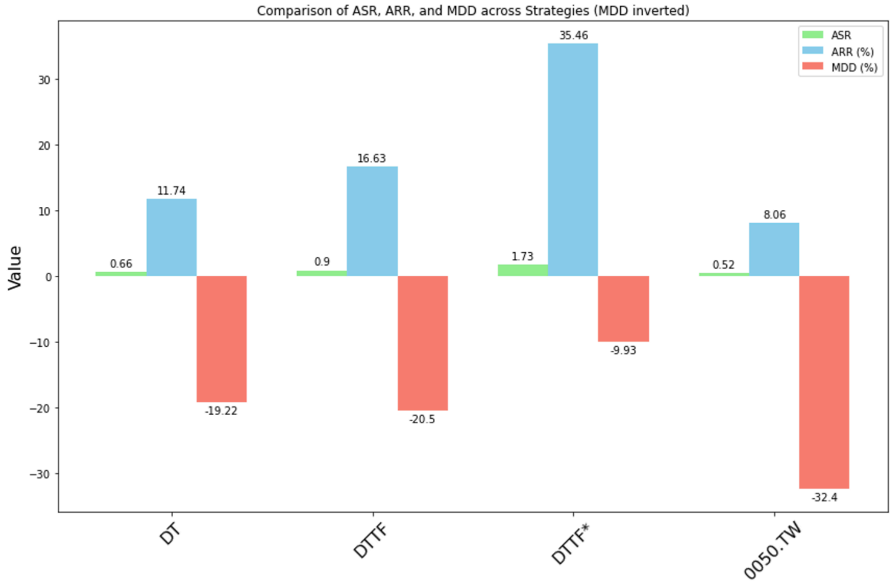

Figure 5.20: Metrics values comparative of Exp4

5.4.3 Comparative Analysis

Table 5.19, Figures 5.21 and 5.23 show the winning rate and annual return for different strategies. The Transformer-based model with monthly updates consistently achieved the highest winning rate and annual returns, indicating its superior performance over static models:

Table 5.19: The winning rate of different strategies compared to the market at different frequencies of Exp4

	Per trade	**3 month**	**6 month**	**12 month**
DT	49.04%	44.44%	50.00%	33.33%
DTTF	51.92%	52.78%	55.56%	66.67%
DTTF*	60.58%	69.44%	77.78%	88.89%

Table 5.20 and Figures 5.22 present the yearly return comparison. It is evident that the Transformer-based architecture with monthly updates consistently achieved the highest annual returns across different years:

doi:10.6342/NTU202403155

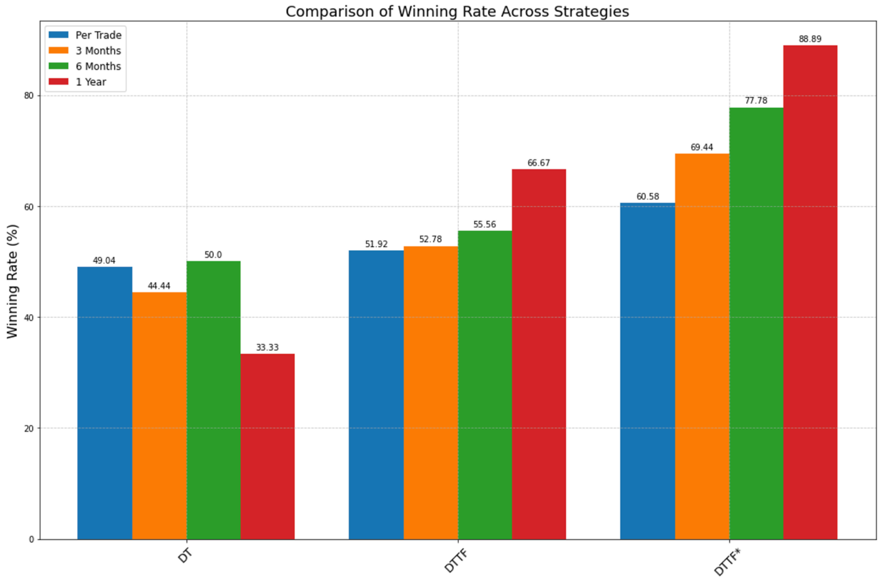

Figure 5.21: Winning rate comparison of Exp4

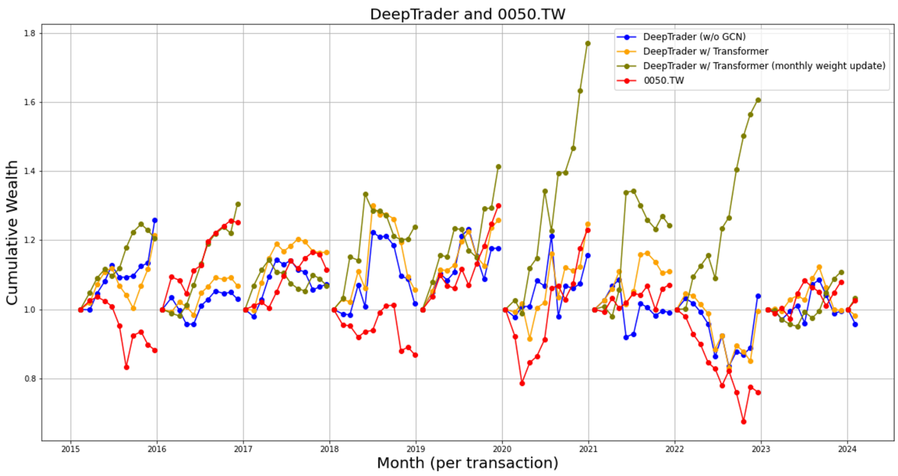

Figure 5.22: Cumulative wealth every year corresponding to each strategy in TW stock market of Exp4

Table 5.20: Annual return of Exp4 from 2015 to 2023

	2015	2016	2017	2018	2019	2020	2021	2022	2023
DT	25.89%	2.97%	7.26%	1.69%	17.77%	15.66%	-0.85%	3.97%	-0.50%
DTTF	21.36%	6.80%	16.52%	5.72%	25.80%	24.75%	11.01%	-0.57%	-0.33%
DTTF*	20.53%	30.55%	6.90%	23.97%	41.35%	77.04%	24.33%	60.57%	10.87%
0050.TW	-11.74%	25.11%	11.53%	-13.04%	30.00%	22.95%	7.15%	-23.86%	8.00%

doi:10.6342/NTU202403155

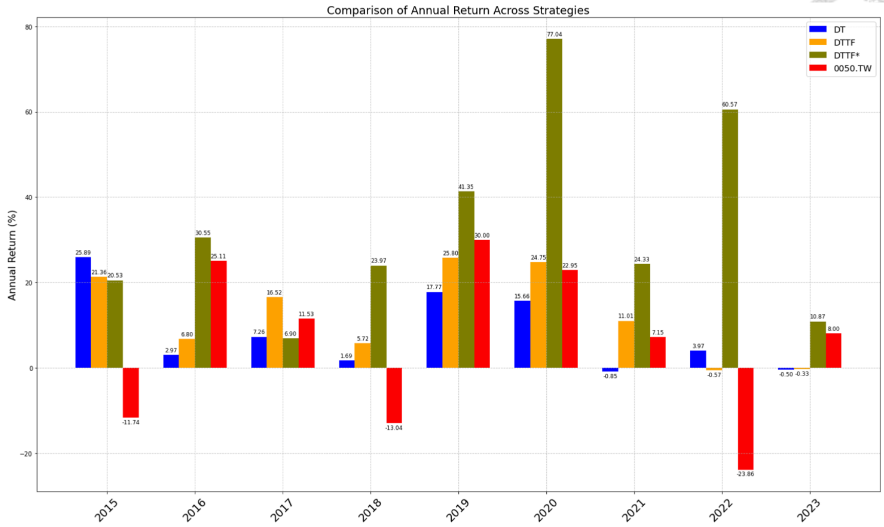

Figure 5.23: Annual return comparison of Exp4

5.4.4 Discussion and Implications

The experiment underscores several critical findings regarding the periodic retraining of the DeepTrader model with updated information:

1. **Enhanced Predictive Accuracy**: Regular updates allowed the model to remain attuned to the latest market trends, significantly improving its predictive capabilities.

2. **Reduced Risk Exposure**: The adaptive approach minimized drawdowns, showcasing the model's ability to manage and mitigate risk effectively.

3. **Improved Consistency**: Continuous updates led to consistent performance improvements, ensuring robust and reliable investment strategies across varying market conditions.

4. **Practical Applicability**: The dynamic retraining strategy aligns with real-world investment practices, where constant adaptation to new information is crucial for maintaining competitive edge and achieving superior returns.

doi:10.6342/NTU202403155

However, it was observed that the original DeepTrader model's performance did not significantly improve with monthly updates. This limitation is attributed to the Graph Convolutional Network (GCN) component, which struggles to focus on specific solutions when dealing with noisy data. To address this, we introduced a Transformer-based attention mechanism to replace the GCN, thereby enhancing the model's stability and adaptability to market changes.

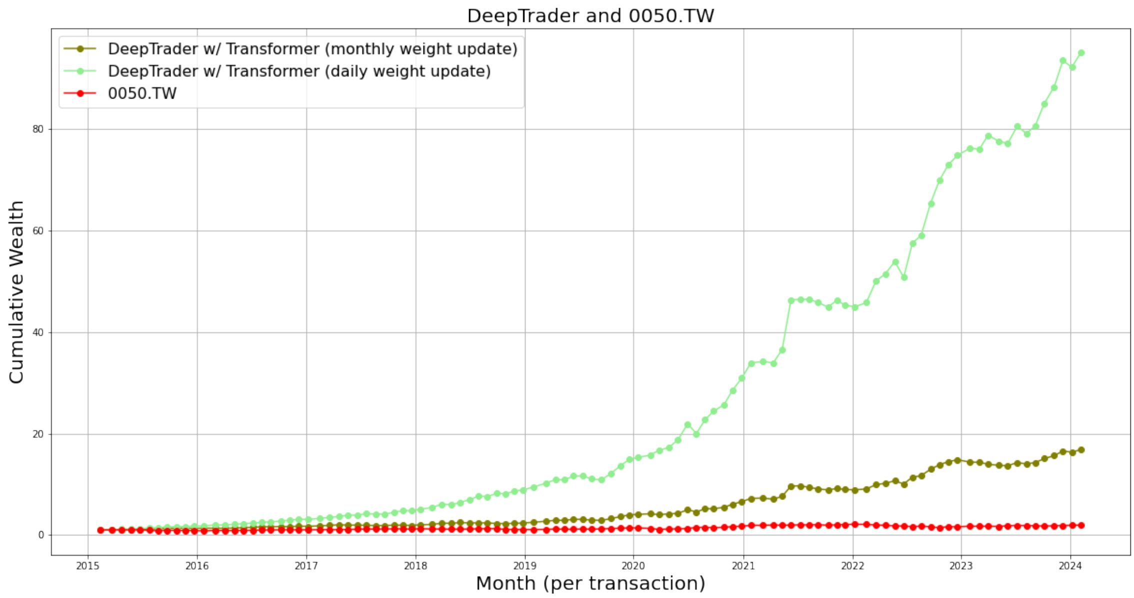

Figure 5.24: Comparison of monthly vs. daily weight adjustments in the new DeepTrader model

Additionally, previous experiments involved making decisions only once a month, which did not fully leverage the model's potential to adapt to rapid market changes. Considering the dynamic nature of financial markets, the model should be capable of adjusting trading frequency as needed. By updating the model to adjust weights daily, we observed significantly higher returns compared to monthly adjustments. This daily trading approach better reflects the market conditions and demonstrates that the Transformer-based attention mechanism allows the model to more effectively find the most suitable strategy for the current market, as illustrated in Figure 5.24.

doi:10.6342/NTU202403155

These conclusions emphasize the necessity of adaptive learning and continuous model refinement in the realm of financial forecasting and investment strategies, ensuring models remain relevant and effective in ever-changing market environments.

doi:10.6342/NTU202403155

Chapter 6 Conclusions and Future Directions

This chapter describes the key findings of the research and offers conclusions based on the experimental results presented in the previous chapter. It summarizes the effectiveness of the proposed models and their contributions to the field of financial trading. Additionally, the chapter outlines potential directions for future research, suggesting areas where further improvements can be made and new methodologies that could be explored to advance financial modeling techniques.

6.1 Conclusions

Based on the experimental findings and analyses conducted with DeepTrader, several key insights and areas for improvement have been identified:

1. **Performance Metrics and Strategy Effectiveness**: DeepTrader demonstrates a higher winning rate and return on short strategies, underscoring its effectiveness in capitalizing on market downturns and volatility spikes. This capability positions DeepTrader as a potent tool for short-term trading strategies, where quick response to market shifts is critical for profitability.

doi:10.6342/NTU202403155

2. **Challenges and Opportunities in the Taiwanese Market**: However, the application of DeepTrader in the Taiwanese market revealed challenges in effectively modeling long-term correlations. This deficiency suggests opportunities for enhancing the model's ability to capture and leverage historical trends over extended periods. Improvements in long-term correlation modeling are crucial for achieving sustained performance and reducing volatility in investment outcomes.

3. **Enhanced Adaptability with Transformer-based Attention**: The integration of Transformer-based attention mechanisms into DeepTrader brought about a substantial improvement in the model's adaptability and overall performance. Transformers are particularly adept at capturing complex dependencies and non-linear relationships within data sequences, which significantly enhances predictive accuracy and robustness across various market conditions. This capability is crucial for navigating unpredictable market dynamics and effectively mitigating risks. The effectiveness of the Transformer architecture was crucial in enabling the model to fully leverage the benefits of continuous weight updates, demonstrating a marked improvement in performance compared to traditional models where such updates had limited impact.

4. **Advantages of Continuous Learning with New Information**: Adopting a continuous retraining strategy for DeepTrader, incorporating the latest market data, proved highly advantageous in maintaining the model's relevance and effectiveness over time. Regularly updating model parameters allowed DeepTrader to stay attuned to evolving trends and shifts in market sentiment, improving its adaptability and consistency in generating profitable investment decisions across different market cycles. The significant gains observed with this approach were largely attributable

doi:10.6342/NTU202403155

to the Transformer-based improvements in the model. Without the enhanced capabilities provided by Transformers, the benefits of frequent weight updates would not have been as pronounced, highlighting the crucial role of Transformer architecture in achieving substantial performance enhancements.

6.2 Future Research Directions

Looking ahead, several strategic directions can further elevate DeepTrader's capabilities and applicability in financial markets:

1. Exploration of Advanced Reinforcement Learning Algorithms: Transitioning from the current Policy Gradient approach to more advanced reinforcement learning algorithms, such as Proximal Policy Optimization (PPO), holds promise for enhancing DeepTrader's decision-making prowess. These algorithms are adept at learning optimal strategies through trial and error, potentially improving the model's ability to navigate complex decision landscapes and optimize investment returns.

2. Integration of Market Sentiment Index as a Predictive Feature: Incorporating the market sentiment index as a feature within DeepTrader's Market Scoring Unit (MSU) could provide deeper insights into investor sentiment and emotional factors influencing market dynamics. Sentiment indicators derived from social media, news sentiment analysis, or investor sentiment surveys could serve as valuable inputs for refining the model's predictions and risk assessments. For example, positive sentiment trends might indicate bullish market behavior, prompting the model to adjust risk exposure or trading strategies accordingly.

doi:10.6342/NTU202403155

doi:10.6342/NTU202403155

References

[1] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[2] Qingnan Sun, Marko V Jankovic, Lia Bally, and Stavroula G Mougiakakou. Predicting blood glucose with an lstm and bi-lstm based deep neural network. In 2018 14th symposium on neural networks and applications (NEUREL), pages 1–5. IEEE, 2018.

[3] Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. Deeptrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 643–650, 2021.

[4] Harry Markowitz. Portfolio selection. The Journal of Finance, 7(1):77–91, 1952.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[6] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. IEEE transactions on neural networks and learning systems, 28(3):653–664, 2016.

doi:10.6342/NTU202403155

[7] John Moody, Matthew Saffell, Yuansong Liao, and Lizhong Wu. Reinforcement learning for trading systems and portfolios: Immediate vs future rewards. In Decision Technologies for Computational Finance: Proceedings of the fifth International Conference Computational Finance, pages 129–140. Springer, 1998.

[8] Financial Times. Jpmorgan develops robot to execute trades, 2017. Accessed: 2024-08-06.

[9] PYMNTS. Ai explained: Reinforcement learning and how it shapes commerce, 2024. Accessed: 2024-08-06.

[10] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059, 2017.

[11] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 1900–1908, 2019.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[13] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 11106–11115, 2021.

doi:10.6342/NTU202403155

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[15] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In Proceedings of the 27th ACM international conference on information and knowledge management, pages 1655–1658, 2018.

[16] Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. Temporal and heterogeneous graph neural network for financial time series prediction. In Proceedings of the 31st ACM international conference on information & knowledge management, pages 3584–3593, 2022.

[17] Kaijian He, Qian Yang, Lei Ji, Jingcheng Pan, and Yingchao Zou. Financial time series forecasting with the deep learning ensemble model. Mathematics, 11(4):1054, 2023.

[18] Khaled A Althelaya, El-Sayed M El-Alfy, and Salahadin Mohammed. Stock market forecast using multivariate analysis with bidirectional and stacked (lstm, gru). In 2018 21st Saudi Computer Society National Computer Conference (NCC), pages 1–7. IEEE, 2018.

[19] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, 90:106181, 2020.

[20] Xiao-Yang Liu, Zhuoran Xiong, Shan Zhong, Hongyang Yang, and Anwar Walid. Practical deep reinforcement learning approach for stock trading. arXiv preprint arXiv:1811.07522, 2018.

doi:10.6342/NTU202403155

[21] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. arXiv preprint arXiv:2011.09607, 2020.

[22] Musonda Katongo and Ritabrata Bhattacharyya. The use of deep reinforcement learning in tactical asset allocation. Available at SSRN 3812609, 2021.

[23] Damian Kisiel and Denise Gorse. Portfolio transformer for attention-based asset allocation. In International Conference on Artificial Intelligence and Soft Computing, pages 61–71. Springer, 2022.

[24] Xiaokang Hu. Stock price prediction based on temporal fusion transformer. In 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), pages 60–66. IEEE, 2021.

[25] Zura Kakushadze. 101 formulaic alphas. Wilmott, 2016(84):72–81, 2016.

doi:10.6342/NTU202403155