

AI Prediction for Stock Trading

ChengLin Yi

Submitted to Auckland University of Technology in partial fulfillment of the requirements for the degree of Master of Computer and Information Sciences (MCIS).

2025

School of Engineering, Computer and Mathematical Sciences

Declaration

I solemnly affirm that, to the best of my understanding and judgment, this thesis is my original work and has not been published or written elsewhere. Unless specifically indicated, it has been completed without external assistance. This submission is presented to Auckland University of Technology as part of the requirements for the Master of Computer and Information Sciences degree. It has neither been submitted previously nor presented to any other academic institution for a qualification.

Signature:

Date: 26 May 2025

Abstract

The key element of stock trading is accurate stock price prediction, and reliable and stable decisions can significantly affect profitability. Traditional prediction models have limitations due to intricacies and instantaneous dynamics within financial environments. This study explores an Artificial Intelligence (AI)-driven stock price prediction method tailored for stocks trading, using advanced machine-learning and deep-learning models to improve forecast accuracy and trading efficiency.

An initial focus was placed on the key role of feature engineering and selection in optimizing model performance. It integrates a variety of technical indicators such as moving averages (SMA/EMA), relative strength index (RSI), MACD and Bollinger bands, as well as principal analysis (PCA) and quantum-inspired feature extraction, and scaling methods such as MinMaxScaler are used to standardized input data. These are combined with advanced feature filtering techniques to enhance overall model performance.

The research integrates different AI methods, feature engineering with historical price trends, market indicators, and sentiment insights drawn from financial news and social platforms, Natural Language Processing (NLP) technology is included to enhance the prediction model by capturing market sentiment and investor behavior. These include Long Short-Term Memory (LSTM) networks, transformer models, LightGBM, Reinforcement Learning (RL), and Hybrid LSTM + Transformer with Quantum-Enhanced Feature Selection.

To evaluate the true effectiveness of the AI models, the study conducted rigorous back testing using real-time stock price data and annualized returns using actual simulated investments. The performance of AI-driven models was compared with traditional benchmarks, including ARIMA, GARCH, and classic machine learning algorithms such as support vector machines (SVM) and random forests. The main technical indicators used to evaluate the model are R^2 , mean absolute error (MAE), and root mean square error (RMSE).

The experimental results demonstrate that the AI hybrid model achieves $R^2 = 0.97$ (MAE = 0.0186, RMSE = 0.0278) on NVDA 15-minute intraday data in predicting short-term stock price movements, which is essential for ultra-short-term and ultra-high frequency trading strategies. Model interpretability was further explored using SHAP (which measures the average marginal contribution of each feature) and LIME (which provides instance-level explanations), helping ensure transparency in the AI-driven decision-making process.

In addition, a hybrid data merging strategy and multiple feature engineering data are proposed to increase robustness by integrating a hybrid AI structure of deep learning and traditional statistical models. This study sheds light on how AI can help lower risks associated with stock market investments and optimizing options trading decisions. However, challenges such as market emergencies events, data noise, and overfitting require further exploration.

Overall, this research demonstrates the power of AI models to predict the financial market and provides help for investors and researchers who use AI to make advanced financial forecasts.

Keywords: Stock price prediction, Options trading, Transformer-based models, Quantum computing, Deep learning.

Acknowledgment

First, I extend my heartfelt appreciation to Studylink for its financial support, which enabled me to successfully complete my master's degree at AUT. I am particularly grateful to my supervisor Parma Nand, who not only imparted valuable professional knowledge throughout the learning process but also provided me with great help in academic and personal growth. At the same time, I would also like to thank the administrative and technical team of AUT for their continuous assistance, as well as the school management for their careful guidance and support throughout my study.

ChengLin Yi
Auckland, New Zealand
2025

Table of Contents

Chapter 1 Introduction	1
1.1 Background and Motivation.....	2
1.2 Problem Statement.....	3
1.3 Research Questions.....	4
1.4 Research Objectives.....	5
1.5 Significance of Study.....	5
1.6 Structure Overview.....	6
Chapter 2 Literature Review	8
2.1 Overview of Stock Market Prediction	9
2.1.1 Traditional Methods for Stock Market Prediction.....	9
2.1.2 Statistical and Econometric Models.....	10
2.1.3 Application of machine learning in stock prediction.....	10
2.1.4 Sentiment Analysis in Stock Forecasting.....	11
2.1.5 Predicting the stock market with hybrid AI models.....	12
2.1.6 Limitations and Challenges of Stock Forecasting.....	12
2.1.7 Conclusion.....	13
2.2 Application of machine learning in finance	13
2.2.1 Development of machine learning in finance.....	14
2.2.2 Models in Stock Forecasting.....	14
2.2.2.1 Long Short-Term Memory (LSTM) Networks.....	14
2.2.2.2 Transformer Models in Finance.....	15
2.2.3 Reinforcement Learning in Stock Trading.....	15
2.2.3.1 Q-Learning and Deep Q Networks (DQN).....	15
2.2.3.2 Actor-Critic and Proximal Policy Optimization (PPO).....	16
2.2.4 The Role of Machine Learning in High-Frequency Trading.....	16
2.2.5 Challenges of Machine Learning in Financial Markets.....	16
2.2.6 Conclusion.....	17
2.3 Automated Trading Strategies	17
2.3.1 The development of algorithmic trading.....	17
2.3.2 High-frequency trading (HFT) and market microstructure.....	18
2.3.3 Algorithmic trading strategies based on artificial intelligence.....	18
2.3.3.1 Optimizing trading strategies with reinforcement learning.....	18
2.3.3.2 Application of deep learning in market forecasting.....	19
2.3.3.3 Application of NLP analysis of sentiment in automated trading.....	19
2.3.4 Risk Management in Algorithmic Trading.....	19

2.3.5 Risks and Challenges of Automated Trading	20
2.3.6 Conclusion.....	21
2.4 Challenges in Stock Prediction	21
2.4.1 Market volatility and non-stationarity	21
2.4.2 Data Noise and Signal Extraction.....	22
2.4.3 Challenges of AI Model Overfitting.....	22
2.4.4 Data Scarcity and Quality Issues	23
2.4.5 Model Interpretability and Trust.....	23
2.4.6 Regime Shifts and Black Swan Events	24
2.4.7 Trade-offs Between Complexity and Performance	24
2.4.8 Conclusion.....	25
Chapter 3. Methodology	26
3.1 Data Collection and Preprocessing	27
3.1.1 Data Sources.....	27
3.1.2 Technical indicators and feature extraction.....	29
3.1.3 Processing of missing values and noisy data.....	30
3.1.4 Data alignment and window processing in time series models	32
3.1.5 Feature Engineering Tools and Data Pipelines.....	32
3.1.6 Data segmentation and cross-validation	33
3.1.7 Ethical considerations and data licensing	34
3.2 Feature Engineering	34
3.2.1 Rolling Indicators and Volatility Features.....	35
3.2.2 Multi-scale moving average and trend identification indicators	36
3.2.3 Volatility index and risk feature extraction	37
3.2.4 Momentum Features.....	39
3.2.5 Sentiment and NLP-derived features.....	41
3.2.6 Quantum-inspired feature selection (QIFS).....	42
3.2.7 Final feature set and dimension construction	44
3.3 AI models used.....	45
3.3.1 Long Short-Term Memory Network (LSTM).....	45
3.3.2 Transformer architecture	48
3.3.3 Reinforcement Learning (RL) for Trading Decisions	50
3.3.4 LightGBM for feature importance and ensemble prediction.....	52
3.3.5 Hybrid LSTM + Transformer with Quantum-Enhanced Feature Selection	53
3.3.6 Model Interpretability Based on SHAP.....	55
3.3.7 Evaluation Summary	56
3.4 Conclusion	56

Chapter 4 Results and Analysis	57
4.1 Trading Strategy Design	58
4.1.1 Signal Generation Framework.....	58
4.1.2 Entry and Exit Criteria.....	58
4.1.3 Position Sizing and Capital Allocation.....	59
4.1.4 Strategy Variants	60
4.1.5 Risk Management Protocol	61
4.1.6 Strategy Evaluation and Deployment	61
4.2 Backtesting Framework	63
4.2.1 Backtesting Environment Setup	63
4.2.2 Dataset and Timeframe.....	64
4.2.3 Evaluation Metrics.....	65
4.2.4 Strategy Comparison and Walk-Forward Analysis	66
4.2.5 Robustness Testing and Real-World Comparison.....	66
4.3 Model Performance	67
4.3.1 Evaluation Indicators.....	67
4.3.2 Model setup and evaluation environment.....	68
4.3.3 Results by Forecast Horizon.....	69
4.3.4 Model Behavior and Residual Analysis	69
4.3.5 Robustness and Summary of Findings	70
4.4 Trading Strategy Evaluation	70
4.4.1 Evaluated Strategies and Performance Results.....	71
4.4.2 Risk-Adjusted Performance.....	72
4.4.3 Drawdown and Volatility Analysis	72
4.4.4 Trade Execution Metrics	73
4.4.5 Performance During Market Conditions	73
4.4.6 Statistical Significance Testing	74
4.4.7 Interpretability of Trading Outcomes	74
4.4.8 Intraday Options Forecast Case Study.....	74
4.5 Comparison of Models	77
4.5.1 Predictive Accuracy Comparison	78
4.5.2 Profitability and risk-adjusted returns	78
4.5.3 Comparative analysis of model characteristics.....	78
4.5.4 Justification for Superior Performance of the Hybrid Model	79
4.5.5 Conclusion.....	79
4.6 Volatility and Risk Management	80
4.6.1 Impact of Volatility on Model Performance.....	80

4.6.2 Volatility-Aware Feature Engineering	80
4.6.3 Dynamic Position Sizing and Volatility Scaling	81
4.6.4 Stop-Loss and Trailing Protection	81
4.6.5 Strategy Suspension Protocols.....	81
4.6.6 Quantitative Risk Metrics.....	81
4.6.7 Stress Testing and Scenario Analysis.....	82
4.6.8 Risk Management Summary	82
4.6.9 Summary.....	83
Chapter 5 Discussion.....	84
5.1 Interpretation of Results	85
5.1.1 Explainability and Practical Value	85
5.1.2 Implications for Trading Systems.....	84
5.2 Challenges Faced.....	86
5.2.1 Sentiment Signal Delay and Noise	86
5.2.2 Model Overfitting and Regularization.....	86
5.2.3 Feature Alignment and Window Synchronization	87
5.2.4 Market Regime Shifts and Anomalies.....	87
5.2.5 Strategy Execution and Latency	87
5.2.6 Model Interpretability and Trust.....	88
5.3 Comparison with Previous Studies	88
5.3.1 Alignment with Prior Studies	89
5.3.2 Advancing the Use of Sentiment and Quantum-Inspired Features.....	89
5.3.3 Reinforcement Learning: A Promising but Underperforming Tool	90
5.3.4 Novel Contributions of This Study.....	90
5.3.5 Limitations in Prior Work Addressed.....	90
5.3.6 Summary.....	91
Chapter 6 Conclusion.....	92
6.1 Summary of Findings	93
6.2 Contributions to FinTech.....	94
6.2.1 Advanced Hybrid AI Framework	94
6.2.2 Real-Time Sentiment Integration	94
6.2.3 Bridging AI with Options Trading	94
6.2.4 Explainability and Trustworthiness	94
6.2.5 Deployment-Ready Infrastructure	95
6.3 Future Work	95
6.3.1 Improve the Generalization and stability of the model	95

6.3.2 Multi-Asset Forecasting and Strategy Generalization.....	95
6.3.3 Expanding into Derivatives and Portfolio Optimization	96
6.3.4 Regulatory and Ethical Considerations	96
6.3.5 Hardware Acceleration and Quantum Integration.....	96
References.....	97

List of Figures

Figure 3.1 Tweet sentiment distribution	29
Figure 3.2 Boxplots for key stock features	31
Figure 3.3 Sliding window approach for time series forecasting	32
Figure 3.4 Comparison of forecast errors between the test set and the validation set over 15 forecast days	34
Figure 3.5 Close price with EMA10 and EMA30 for trend detection.....	36
Figure 3.6 MACD and signal line crossover for NVDA.....	37
Figure 3.7 Bollinger bands (± 2 Std Dev) and close price.....	38
Figure 3.8 Visual Comparison of Momentum Indicators.....	40
Figure 3.9a Daily Polarity and Subjectivity Trends.....	42
Figure 3.9b Daily Hashtag Trend - bullish vs #bearish.....	42
Figure 3.10 Quantum-Inspired Feature Score Ranking.....	44
Figure 3.11 MAE and RMSE by forecast day (LSTM).....	48
Figure 3.12 Transformer forecast error by forecast day.....	50
Figure 3.13 DQN Agent Training - Total Reward per Episode.....	51
Figure 3.14 Epsilon Decay Over Training Episodes.	52
Figure 3.15 LightGBM Forecast Error by Day (Return).....	53
Figure 3.16 MAE and RMSE per forecast day (quantum-enhanced hybrid model) ...	55
Figure 3.17 SHAP feature importance (Top 10) – Quantum hybrid model.....	55
Figure 4.1 Signal generation logic flowchart.....	59
Figure 4.2 AI Prediction Stock for Trading Flowchart)	62
Figure 4.3 NVDA stock real-time price prediction for trading day (April 14, 2025)	65
Figure 4.4 Actual vs Predicted NVDA Prices in Intraday Forecast Simulation (April 14,2025)	75
Figure 4.5 Example of real-time call signal confirmation and trigger using 2-second interval logic.....	76
Figure 4.6 Forecasted NVDA intraday price path with signal zones and option trade triggers (April 14,	

List of Tables

Table 3.1 Yahoo Finance’s raw NVDA stock data sample (2020-2025)	27
Table 3.2 Sentiment related to the 2022 Stock Market Crash.....	28
Table 3.3 Summary of engineered rolling and momentum features.....	35
Table 3.4 Summary of Trend Identification Indicators.....	37
Table 3.5 Volatility indicators summary.....	38
Table 3.6 Momentum indicators summary.....	40
Table 3.7 Summarizing NLP-derived features.....	41
Table 3.8 Top QIFS features with descriptions.....	43
Table 3.9 Structure of Final Model Input Features.....	44
Table 3.10 LSTM forecast error performance for each day (day 1 to 15)	47
Table 3.11 Average forecasting error metrics by horizon (LSTM).....	48
Table 3.12 Enhanced Transformer 15-Day Forecast Accuracy.....	50
Table 3.13 Key Components of the RL trading agent.....	51
Table 3.14 LightGBM multi-step forecasting performance (return).....	53
Table 3.15 Quantum-enhanced hybrid model forecast accuracy (day 1–15)	54
Table 3.16 Provides the corresponding SHAP value rankings of those features.....	56
Table 3.17 Evaluation Summary value rankings of those features.....	56
Table 4.1 AI Model Signal Generation Summary.....	58
Table 4.2 Trading Entry and Exit Criteria Summary.....	59
Table 4.3 Position Sizing and Trade Actions Based on Market Conditions... ..	60
Table 4.4 Summary of Trading Strategy Variants	61
Table 4.5 Strategy Evaluation and Deployment Summary.....	62
Table 4.6 Backtesting Environment Parameters.....	63
Table 4.7 LightGBM Intraday Forecast with Rolling R ² (April 14, 2025)	64
Table 4.8 Monte Carlo Simulation Results (1000 Runs)	67
Table 4.9 Performance summary for 1 day forecast.....	68

Table 4.10 Performance summary for 15-day forecast.....	69
Table 4.11 Strategy descriptions and Performance Metrics.....	71
Table 4.12 Net Profit (USD).....	72
Table 4.13 Risk-Adjusted Performance Metrics.....	72
Table 4.14 Drawdown and Volatility Metrics.....	73
Table 4.15 Trade Execution Metrics.....	73
Table 4.16 Predictive Accuracy Metrics	78
Table 4.17 Profitability metrics across four AI trading models.....	78
Table 4.18 Summarizes the comparative performance of the four models in terms of robustness, computational efficiency, interpretability, and limitations	79
Table 4.19 Market Events, Volatility Index Levels, and Most Affected Models	80
Table 4.20 Quantitative Risk Metrics Under Volatility Scenarios	82
Table 5.1 Summary of Challenges and Mitigation Strategies	88
Table 5.2 Key novel contributions of this study compared to prior work... ..	90
Table 5.3 Key limitations in prior research and how this study addressed them.....	91

Chapter 1

Introduction

This chapter contains six key segments. The first section introduces stock trading and examines the role of AI in financial markets. The second part discusses the challenges involved in forecasting stock prices and trading. After that, the research questions, and objectives are outlined, the significance of the study is explained, and the chapter concludes with an overview of the report's structure.

1.1 Background and Motivation

Since the late 18th century, stock trading has been one of the core pillars of the global financial market. Stocks are an important mechanism for capital allocation, investment growth and economic stability. As an important part of the capital market, stock trading promotes capital flow, thereby driving economic growth, resource allocation and corporate development. Whether it is capital operation or stock market investors, the goal is to maximize investment returns while managing risks. Current trading strategies have evolved from traditional free trading based on basic technical analysis to highly complex algorithm-driven model predictions that leverage the power of AI and data science. As stock markets have become increasingly complex over time, coupled with the increase in data availability and advances in computing power, the use of AI-driven trading strategies has been driven.

Traditional stock price prediction methods (such as observing candlestick charts, real-time open stock prices rising above the previous trading day's closing price, statistics and econometric models) have always been difficult to cope with the dynamic and nonlinear characteristics of financial markets. These methods and models are usually unable to capture the impact of external factors, including stock market sentiment, economic indicators, geopolitical events and industry emergencies. However, AI has become a transformative technology in the financial field. It uses statistics and algorithms for data feature engineering while providing machine learning (ML) together with deep learning (DL) algorithms to improve prediction accuracy and trading efficiency.

With the development of Quantitative Trading and High-Frequency Trading (HFT) trading, AI-driven models are now able to quickly (dependent on GPUs) process large amounts of real-time market data, extract meaningful insights through machine learning, and make split-second trading decisions. AI-driven stock price prediction uses historical data, macroeconomic indicators, sentiment analysis, statistical feature-engineering methods, and technical indicators to generate more accurate predictions. Deep learning hybrid architectures such as LSTM networks, Transformer models, RL, and LightGBM have shown extraordinary promises in stock price prediction.

The stock price prediction trading system based on advanced AI is not only an investment tool for the stock market, but also greatly improves the decision-making ability of market investors, providing a more efficient, accurate and favorable investment method, thus completely changing the option trading strategy. Options trading involves derivative option purchase and sale contracts, whose value depends on the price change of the underlying stock, It shows that when the implied volatility of the underlying asset increases by +1% or -1%, the option value may change significantly due to its 100x leverage value. Therefore, accurate forecasts of stock prices are very important for reducing risks and optimizing investment returns in options trading. Therefore, accurate prediction of stock prices is very important for reducing risks, and optimizing investment returns in options trading. Because of the complexity of option pricing, coupled with the demand for high-frequency and high-speed trading, AI has become a game-leader in the stock market.

1.2 Problem Statement

In the financial market, because of the randomness, nonlinearity, high-frequency fluctuations and intricate nature of the financial market, predicting stock price trends has always been a complex and difficult problem. Although conventional models such as ARIMA (Autoregressive Integrated Moving Average) and GARCH (Generalized Autoregressive Conditional Heteroskedasticity) have been widely used in stock volatility prediction, the stock market is usually uncertain and variable. So, they have major limitations:

- Lack of nonlinear dependence: The price of a stock is affected by many interdependent variables, many of which exhibit nonlinear relationships, making it impossible to model them using traditional statistical methods fully.
- Sensitivity to changes in market conditions: Stock prices are often very sensitive to time factors, and static models become less effective over time for structural changes in financial markets due to economic crises, policy shifts, and technological disruptions.
- Cannot perform feature engineering on unstructured data: Traditional forecasting models mainly rely on numerical time series data, while AI models can form hybrid models based on the different performance of different models, and can combine different data analyses, such as combining earnings reports of issuing companies, industry emergencies and social media sentiment analysis to provide a more comprehensive view of market behavior from multiple perspectives.

Despite the advancements in the application of AI in financial market forecasting and stock trading, there are still some challenges in stock price prediction and Quantitative Trading:

- Volatility and randomness of stock market dynamics: Driven by sudden changes in macroeconomic policies, the unpredictability of geopolitical conflicts, and investor sentiment, stock prices fluctuate suddenly, making it difficult for traditional models to be promoted under different market conditions.
- Overfitting vs. underfitting of AI models: When a model contains too many parameters or is too complex, it tends to react to tiny details in the training data, resulting in overfitting, and underfitting occurs when there is still room for improvement on the test data, and fails to recognize meaningful patterns, resulting in poor generalization to new data.
- Data noise and feature selection: Selecting too few or irrelevant features in a financial dataset may result in the model failing to capture the key information of the data. Selecting too many features may introduce noise and lead to overfitting, so identifying important predictive factors is crucial.
- Explainability of AI-driven predictions: For complex and large AI models, especially the complex architecture of deep learning, they operate in a black box system mode, which makes it difficult for stock traders to achieve clear and effective communication, understanding and trust model-driven decisions.

In high-frequency trading of options, these challenges become even more crucial. Since an option's value is directly linked to the asset's underlying stock price, even minor inaccuracies in predictions can lead to substantial

financial losses. In some cases, this may result in the complete loss of capital invested if the market moves against the option holder. Accurate predictions of real-time stock price movements help option traders make informed decisions about fast execution price trades, expiration dates, and hedging strategies. Therefore, the need for powerful, AI-driven stock forecasting models that enhance option trading decisions and risk management is more important than ever.

1.3 Research Questions

As the volatility and complexity of financial markets continue to increase, traditional stock price prediction methods have always had bottlenecks in accuracy and stability, especially in actual application scenarios targeting automatic trading systems. However, in recent times, with the rapid development of AI, especially the rise of deep learning, Transformer architecture, and multi-model fusion methods, more possibilities have been provided for improving the prediction accuracy of options trading and realizing trading automation.

Therefore, this study will address the following core research questions:

"How accurate can AI prediction models represented by Transformer architecture and RL algorithms be in predicting stock price trends? To what extent can these models effectively optimize automated stock or option trading decisions and actual profit performance?"

To explore this issue in more depth, this study will further focus on the following aspects:

- Compared with traditional machine learning methods (such as LSTM models or random forest algorithms), can AI models based on Transformer architecture show significant advantages in actual stock price prediction? In the prediction of NVDA's stock price in the next 15 days, can the Transformer model combined with quantum computing methods improve the prediction effect?
- To what extent does social media sentiment, especially positive or negative sentiment from Twitter, affect the price trend of a specific stock (such as NVDA)?
- In the actual intraday options trading environment, can RL algorithms effectively transform stock price prediction results into stable and profitable trading strategies?
- When the market encounters extreme, unpredictable market fluctuations or sudden abnormal situations such as Nvidia CEO Jensen Huang's prediction of the prospects of quantum computing, how robust and adaptable are these AI-driven models?

The above research aims to deeply explore the actual performance of the most cutting-edge AI technologies in automated options trading, clarify the advantages, potential risks and application limitations of these technologies, and thus provide meaningful guidance and improvement directions for the development of future financial technology systems.

1.4 Research Objectives

The purpose focused on this study is to explore and evaluate an AI-driven stock price prediction model specifically for common stock real-time stock options trading (e.g., Intraday Options Trading Price Guidance, Real-time stock price prediction every 15 minutes and forecasting stock price trends for the future 15 days). The study aims to:

- Develop an AI-based forecasting hybrid model using LSTM, Transformers, deep learning, and LightGBM to improve forecasting accuracy.
- Enhance feature engineering by incorporating multiple technical indicators, including enhancing stock market datasets by adding technical indicators (Quantum Feature Selection, MACD, RSI, Compute Bollinger Bands), mathematical calculations based on price, volume, or open interest. These indicators help traders analyze price trends and make better trading decisions.
- AI-driven hybrid models are compared with traditional forecasting techniques such as ARIMA models, GARCH, and classic machine learning algorithms that can capture a certain degree of time series characteristics but cannot effectively handle nonlinear changes in a noisy environment, such as support vector machines (SVMs) along with random forests.
- Use interpretability tools, like SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-Independent Explanations) to ensure the model can interpret individual predictions, showing which features were changed to change the prediction.
- AI-based trading strategies are not only backtested on real-time market data, but actual investments are used for proof of concept to assess profitability effectiveness, stability, and risk-adjusted returns.
- Develop a hybrid AI framework that combines hybrid feature engineering, deep learning and multiple models to improve the robustness of market price trend forecasts and adaptability to changes in market mechanisms, thereby achieving the goal of rapid investment profitability.

1.5 Significance of Study

The application of AI in the stock market, especially in professional investment stock trading and options trading, is extremely likely to change investment strategies completely and greatly improve trading frequency and efficiency while minimizing financial risks and improving investment returns. The significance of this research is:

- Improve investment returns: AI models can quickly obtain and analyze large amounts of real-time data, use GPUs to calculate and identify profitable trading opportunities in seconds, and execute high-frequency trading with minimal latency.
- Risk management strategy optimization: Real-time stock price prediction every 15 minutes for intraday options trading and price trends tracking strategies can be used when trading in various time frames.

However, the most accurate predictions and lower risks are associated with Intraday Options Trading Price Guidance. Ensure that losses are minimized.

- Avoiding human bias in trading: In the process of options trading, the use of AI-driven models eliminates emotional decision-making and human cognitive bias, ensuring data-driven systematicity and effective trading strategies.
- Automated UHF Trading Strategies: AI triggers complex trading strategies at ultra-fast speeds by obtaining dynamic real-time price fluctuations 120 times per minute, reducing manual errors and business expenses, improving efficiency and new opportunities for market participants.

This research adds to the convergence of AI, financial technology (FinTech) and quantitative trading, and AI can continuously optimize these models and algorithms through self-learning and adaptation, which not only improves the efficiency and accuracy of transactions, but also reduces transaction risks, bringing new opportunities to the financial market, and provides valuable insights on how to effectively apply AI models to stock price prediction and option trading strategies.

1.6 Structure Overview

This research document outlines the structure as below:

- Chapter 1: Introduction – Overview of stock trading and the application of AI in financial markets, clarifies the research questions, objectives, significance, and sorts out the paper structure.
- Chapter 2: Literature Review - Compile some of the latest research papers on stock trend prediction tasks, which deeply explore existing research on AI-driven stock price prediction models, and analyze their methods, advantages, and limitations.
- Chapter 3: Research Methodology – Describes the data processing and feature engineering, feature selection techniques, AI architectures and optimization, and evaluation metrics used in the study.
- Chapter 4: Experimental Results and Analysis - Presents whether the research methods based on AI models achieve acceptable accuracy results and compares their prediction accuracy and real-time trading performance with traditional benchmarks.
- Chapter 5: Discussion and Conclusion - Summarize key findings and arguments, restate the research questions, discuss challenges, and propose future research directions.
- Chapter 6: Conclusion and Future Research – Summarizes the contribution of this research to financial technology, especially its impact on AI stock trading systems, and proposes possible directions for subsequent research.

By systematically studying the prediction and decision-making methods of financial markets driven by AI, mainly applied to the prediction of stock prices in options trading, this study aims to promote the investment application of AI in the financial market and demonstrate how AI-driven models can optimize trading strategies,

improve risk management and enhance financial decision-making.

Chapter 2

Literature Review

This chapter focuses on a literature review of stock market forecasting methods. It explores four main areas: existing research on stock market forecasting techniques, the application of machine learning in finance to stock trading, previous research on algorithmic trading and the role of AI in trading automation, and issues such as market volatility, noise, and overfitting in AI models.

2.1 Overview of Stock Market Prediction

Stock market forecasts can provide information about future market trends, which have a significant impact on investors (Penman, 2013; Murphy, 1999). Therefore, stock price forecasting has been of long-term interest to economists, investors, and researchers. Accurate predictions of stock price movements can help traders optimize investment strategies in volatile markets, obtain higher returns, and reduce risks. Since the internationalization of financial markets, people have developed various techniques to improve stock price forecasts using a variety of methods such as the principles of mathematical analysis. However, unlike physics or other sciences, there are no basic laws in finance that can derive models (such as Newton's laws). Stock price forecasting technology has evolved from traditional mathematical statistical methods and linear models to more complex deep learning and neural networks. These methods provide investors with different market analysis tools to help them make more reasonable investment decisions. This section will review the existing stock market forecasting methods, focusing on their development process, application effects and limitations.

2.1.1 Traditional Methods for Stock Market Prediction

Initially, stock price predictions relied mainly on the numerical characteristics of historical data and used traditional mathematical algorithms, mainly based on statistics and econometrics, such as moving averages, regression analysis, etc. for analysis. Used to determine the intrinsic value of a stock and compare it with the market price to identify undervalued or overvalued assets. These data analyses include evaluating a company's financial condition and market position by analyzing and evaluating balance sheets, earnings reports, macroeconomic indicators, and industry trends (Penman, 2013).

- In addition, technical analysis mainly observes the historical price movements, open and close prices, trading volume along with various technical indicators of stocks, and the goal is to predict future price trends through pattern recognition (Murphy, 1999). This method relies on market behaviors and statistics rather than the company's financial data. Technical analysis methods include Moving Averages (SMA & EMA): By taking the average of the closing prices of the past N days, it is used to calculate the average of stock prices to smooth price fluctuations and identify market trends.
- Moving Averages (SMA & EMA): By taking the average of the closing prices of the past N days, it is used to calculate the average of stock prices to smooth price fluctuations and identify market trends.
- Relative Strength Index (RSI): It mainly calculates the average increase in the recent rising days or the average decrease in the recent fall days. The speed and change of price changes, as well as indicating the buying and selling strength of the stock, are used to determine whether it is overbought or oversold.
- Bollinger Bands: Use the 20-day simple moving average price fluctuation range to determine the potential price trading status of the market.

- **Candlestick Patterns:** Analyze past price trends through candlestick patterns to predict future price changes.

Although these technical analysis methods are widely used, they also have certain limitations. Fundamental analysis requires investors to have in-depth financial market expertise, and the analysis process may be affected by personal bias and has certain subjective factors. Technical analysis mainly relies on historical data. When the market fluctuates violently or unexpected events occur, the prediction effect may be affected, and it is difficult to accurately predict the future trend of stock prices.

2.1.2 Statistical and Econometric Models

To overcome the limitations of traditional methods, researchers have developed statistical methods and econometric models for stock price forecasting. They reveal the quantitative change law of the economic system more deeply. Some of the most used models include:

- The Autoregressive Integrated Moving Average (ARIMA) model is a statistical technique that captures trends in financial data and is widely used for sequence forecasting (Tsay, 2010).
- Generalized Autoregressive Conditional Heteroskedasticity (GARCH): is a statistical model used primarily in econometrics and finance to analyze time series data that exhibits volatility clustering, Models the clustering of volatility in financial markets to estimate future price volatility (Tsay, 2010).
- Hidden Markov Model (HMM): This statistical framework describes a Markov process characterized by hidden, unknown parameters. It captures the probabilistic nature of stock price changes by modelling market conditions (Hassan & Nath, 2005).

Although these models have improved forecasting performance, they still have difficulty dealing with nonlinear relationships, sudden market changes, and external factors such as geopolitical events and investor sentiment (Tsay, 2010; Hassan & Nath, 2005).

2.1.3 Application of machine learning in stock prediction

Machine learning (ML) and deep learning, unlike traditional methods, ML enables models to learn complex data structure patterns from large data sets quickly (Patel et al., 2015). Widely used to predict stock market trends and prices, ML models can not only adapt to new and changing market environments and conditions but also merge multiple data sources. Some machine learning techniques applied in financial markets include:

- **Decision tree and linear regression:** A decision tree is a complex model that divides data points into multiple different categories by recursively dividing the feature space and is used to predict the value of categorical variables. Linear regression is a simple model that fits data points to get a straight line or curve, which is used to predict the value of continuous variables for price trend prediction (Patel et al., 2015).

- Random forest and gradient boosting (XGBoost, LightGBM) all belong to ensemble learning. Ensemble learning improves the generalization ability of basic learners by aggregating several decision trees to enhance forecasting results (Fischer & Krauss, 2018).
- Support vector machine (SVM): Applicable to binary classification problems and can also be extended to handle multi-classification problems. Classify stock price trends by finding the best decision boundary in high-dimensional data (Fischer & Krauss, 2018).
- Neural network (MLP, CNN): MLP is mainly used to process structured data, while CNN is mainly used to process data with spatial structure to capture the complex relationship between stock prices and multiple financial indicators (Fischer & Krauss, 2018).

The application of deep learning models, such as LSTM and Transformers has further enhanced the ability of ML techniques to predict stock prices (Fischer & Krauss, 2018). Transformers can process sequences in parallel, especially for different parts of long sequences, which greatly reduces training time. These models are good at analyzing temporal data sequences, identifying extended dependencies over time, and improving the accuracy of stock price prediction under dynamic market conditions (Patel et al., 2015).

2.1.4 Sentiment Analysis in Stock Forecasting

Research on real-world financial markets has highlighted the integration of alternative data sources to improve the precision of stock market forecasts. Sentiment analysis techniques powered by NLP enable AI models to extract insights from the following sources (Preis, T., Moat, H. S., & Stanley, H. E. 2013):

- Industry financial news articles and record quarterly revenue reports.
- Social media platforms such as StockTwits, Twitter, and Reddit.
- Investor sentiment indicators from online influential forums.

Research shows that the accuracy of stock forecasting can be significantly improved by capturing investor sentiment analysis and market perception through influential social media platforms (Bollen, J., Mao, H., & Zeng, X. 2011). AI-driven trading systems can process large amounts of unstructured data, classify sentiment polarity, and incorporate this sentiment data into forecasting models.

2.1.5 Predicting the stock market with hybrid AI models

Further, improves prediction accuracy using hybrid AI models that combine multiple techniques. These models combine traditional econometric models with machine deep learning architectures to leverage their respective strengths. Examples include:

- ARIMA and LSTM models: ARIMA models are suitable for extracting short-term stock price trends, while LSTM models can handle nonlinear patterns in time series and perform long-term dependency analysis.
- GARCH and deep learning models: The GARCH model improves volatility prediction by combining statistical methods for stock volatility prediction with neural networks, showing good accuracy (Zhang, 2003).
- RL for trading strategies: The RL model is a machine learning paradigm that focuses on learning the best trading strategy and continuously improving decision-making in a specific environment by interacting with the market environment.

Because the hybrid model fully exploits the advantages of each model, it provides higher robustness, dynamic adaptability and market volatility, making it a promising direction for future stock prediction research.

2.1.6 Limitations and Challenges of Stock Forecasting

Although stock market forecasting has made significant progress in recent years, this task is still challenging due to the following factors:

- Market noise and volatility: Market noise is price data that distorts the underlying trend graph and is usually characterized by high volatility. Volatility is a measure of the degree of change in asset prices over a specific period. With noisy data, it is difficult for models to distinguish meaningful patterns from random fluctuations (Cont, 2001).
- Overfitting of machine learning models: Because the AI model is too complex and performs too well in the training process, or may rely too much on historical data, it performs poorly in the validation data set and the test data set and cannot adapt well to market changes (Srivastava et al., 2014).
- Policy changes and emergencies: An important reason is that the occurrence, development, expansion and possible impact are uncertain, and accompanied by difficulties such as tight decision-making time and information asymmetry. Government policy adjustments and sudden global events (such as financial crises, geopolitical conflicts or influential industry announcements) will have a significant impact on market trends, making it difficult for the model's prediction results to remain stable (Cont, 2001).
- Data quality and feature selection: Screening out valuable data indicators from the big data market is the key to improving prediction accuracy. If the data quality does not meet the requirements of the model or

the feature engineering is not properly selected, the performance of the model may be seriously affected (Srivastava et al., 2014).

- Model interpretability issues: Deep learning models run like a "black box" and predict price trends. The model cannot explain itself, making it difficult for traders to understand the basis for its predictions. This lack of transparency may affect market participants' trust in AI decisions (Ribeiro, Singh, & Guestrin, 2016).

To overcome these realistic problems, it is essential to continuously optimize the adaptive learning model, improve the interpretability of AI and the transparency of the algorithm, and combine more scientific feature engineering methods. It is necessary to combine AI with financial expertise to achieve more reliable and practical real-time stock market predictions.

2.1.7 Conclusion

Stock market prediction technology has undergone a transformation from traditional fundamental analysis and technical analysis to today's reliance on AI models. Although early statistical models have improved the accuracy of market analysis to a certain extent, they have limitations in dealing with nonlinear market changes and complex dynamic environments. With the development of machine learning and deep learning, models such as LSTM, Transformers, and RL have brought stronger adaptability and more sophisticated analysis methods to market prediction. In addition, sentiment analysis and alternative data sources (such as social media, news sentiment, search trends, etc.) provide new perspectives and data support for market prediction.

Nevertheless, stock market prediction still faces many challenges, including unstable data quality, model interpretability issues, and high uncertainty in the market itself. Future research should further explore hybrid AI models, combine real-time data streams, and improve the transparency and understandability of AI predictions to enhance their practicality and credibility.

This section provides an overall review of the development of stock market prediction technology, laying the foundation for subsequent in-depth discussions on the application of machine learning in the financial field, automated trading strategies, and how AI can assist investment decisions.

2.2 Application of machine learning in finance

Machine learning is profoundly changing the financial market, especially in the trading field of the stock market. It enables models to quickly massive amounts of data through GPUs, identify hidden market patterns, and assist investors in generating more accurate trading decisions in real time. ML is significantly different from traditional forecasting methods. It has stronger adaptive capabilities, can cope with rapid and dynamic changes in the market, can quickly optimize portfolio strategies and improves the efficiency of transaction execution. This

section will explore the application of machine learning in stock market trading, focusing on deep learning models (for example, LSTM, Transformers), and RL, and discuss how they affect algorithmic trading strategies, providing investors with smarter market analysis and decision-making tools.

2.2.1 Development of machine learning in finance

In the past, stock market forecasting mainly relied on econometric models, such as the ARIMA for price levels or returns, while the GARCH attempts to model clustering of volatility or squared returns and the hidden Markov model (HMM) (Tsay, 2010). Although these methods can identify market trends and cyclical changes, they are often unable to cope with nonlinear market relationships and complex market dynamics.

With the improvement of computing power and the rapid growth of financial data, machine learning (ML) has gradually become a more advantageous alternative. Early financial market applications mainly used linear regression, decision trees, and support vector machines (SVM) to predict price trends (Patel et al., 2015). However, with the rise of deep learning, traditional methods have gradually been replaced by more advanced neural networks. Deep learning technology can more accurately capture the time series characteristics and potential trading patterns in the market, providing investors with a more forward-looking market analysis (Fischer & Krauss, 2018).

2.2.2 Models in Stock Forecasting

Forecasting stock price trends, deep learning models such as LSTM networks and Transformers significantly improve stock price forecasting as they can handle long-term data and complex dependencies of time series data patterns (Hiransha et al., 2018). These models address key challenges in stock market forecasting, such as market dynamics and nonlinear dependencies.

2.2.2.1 Long Short-Term Memory (LSTM) Networks

LSTM networks are a special type of Recurrent Neural Network (RNN) architecture specifically designed to address the limitations of standard RNNs when processing time series data, especially the vanishing gradient problem. It can effectively filter, store, and update key information by introducing memory cells and gating mechanisms, thereby maintaining relevance in data over a long period. (Hochreiter & Schmidhuber, 1997). Its excellent performance is as follows:

- Capture the long-term trend and market fluctuation of stock prices and avoid information loss in long-term dependence.
- Reduce the error accumulation in sequence prediction and improve the forecasting accuracy of time series data.

- Support multiple market data variable inputs, such as technical indicators, macroeconomic data, and market sentiment, thereby improving the comprehensiveness of predictions. (Fischer & Krauss, 2018).

Studies have shown that the application of LSTM in stock trading is effective. For example, Fischer & Krauss (2018) trained LSTM models on S&P 500 index data and found that their prediction performance was better than traditional methods such as logistic regression and random forests.

2.2.2.2 Transformer Models in Finance

Transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), have been increasingly applied in financial market forecasting (Zhang et al., 2017). These models leverage self-attention mechanisms to analyze a large volumes of historical price data, enabling them to detect intricate patterns that traditional models may overlook (Vaswani et al., 2017).

Key applications of Transformer models in finance include:

- Sentiment Analysis: Evaluating news sentiment and its impact on stock price movements (Xing et al., 2021).
- Unstructured Data Processing: Extracting insights from earnings reports and analyst recommendations.
- Time-Series Forecasting: Enhancing the accuracy of financial asset price predictions (Zubair et al., 2022).

By incorporating Transformers into financial modelling, analysts can gain deeper insights from vast and complex datasets, improving decision-making in stock trading and investment strategies.

2.2.3 Reinforcement Learning in Stock Trading

RL has become an essential tool in algorithmic trading due to its ability to learn optimal trading strategies through continuous interaction with the market environment (Deng et al., 2017). In contrast to conventional supervised learning models, which rely on labeled historical data, RL agents dynamically adjust their trading strategies based on a reward system that considers factors such as profitability and risk management.

By continuously refining their decision-making process, RL-based trading models can adapt to changing market conditions, improve execution efficiency, and optimize portfolio performance, making them highly valuable for modern financial markets.

2.2.3.1 Q-Learning and Deep Q Networks (DQN)

Q-Learning functions as a fundamental RL algorithm in which an agent learns the best trading actions by maximizing cumulative rewards (Mnih et al., 2015). Deep Q Networks (DQN) take this approach further by

integrating deep neural networks to approximate Q-value functions, making them highly effective in financial markets.

With the ability to process complex market patterns, DQN-based models can:

- Execute real-time trading decisions by continuously learning from market fluctuations.
- Optimize buy/sell signals based on historical price trends.
- Minimize risk exposure through adaptive learning strategies (Deng et al., 2017).

By leveraging deep learning, DQN enhances decision-making in stock trading, helping traders navigate market volatility more effectively.

2.2.3.2 Actor-Critic and Proximal Policy Optimization (PPO)

Advanced RL frameworks like Actor-Critic and Proximal Policy Optimization (PPO) have enhanced stock trading strategies by balancing market exploration and exploitation (Schulman et al., 2017). These methods refine key aspects of trading, including:

- Portfolio allocation and diversification, ensuring better risk management.
- Stop-loss and take-profit orders, helping traders manage downside risk effectively.
- Risk-adjusted return optimization, improving long-term profitability in financial markets (Jiang et al., 2021).

By integrating these RL approaches, traders can adapt to changing market conditions, optimize trading performance, and enhance decision-making in algorithmic trading strategies.

2.2.4 The Role of Machine Learning in High-Frequency Trading

High-Frequency Trading (HFT) leverages machine learning algorithms to execute many trades within milliseconds (Kearns & Nevmyvaka, 2013). These ML-driven HFT strategies benefit from:

- Real-time market microstructure analysis, identifying short-term price inefficiencies.
- Automated arbitrage trading, capturing profit opportunities across different markets.
- Latency-sensitive order execution, optimizing trade speed to gain a competitive edge (Kearns & Nevmyvaka, 2013).

By applying advanced ML techniques, HFT firms can enhance trade execution, reduce risk, and maximize profits in highly volatile and competitive markets.

2.2.5 Challenges of Machine Learning in Financial Markets

While machine learning has demonstrated notable success in stock trading, several practical challenges remain:

- **Poor Data Quality:** Financial market data is often noisy, disorganized, and frequently impacted by regulatory or institutional changes (Cont, 2001).
- **Risk of Overfitting:** Models trained extensively on historical data may struggle to perform effectively in new or unforeseen market conditions, limiting their practical applicability (Srivastava et al., 2014).
- **Lack of Interpretability:** Complex machine learning models can function like a "black box," making it difficult for traders to clearly understand how trading decisions are derived (Ribeiro et al., 2016).

2.2.6 Conclusion

Machine learning has significantly transformed stock trading by allowing predictive models to efficiently handle large financial datasets, adapt flexibly to shifting market conditions, and improve algorithmic trading strategies. Techniques like LSTM, Transformer models, and RL have considerably boosted performance in price prediction, risk management, and high-frequency trading. Although much success has been achieved, practical issues such as overfitting, interpretability, market instability, and volatility of market dynamics still pose challenges and require continued research and improvement of AI-driven financial systems. Future advances, especially explainable AI and hybrid AI approaches are expected to further improve automated stock trading and portfolio optimization.

2.3 Automated Trading Strategies

Automated trading, also known as algorithmic trading, is a method of executing trading orders using automated systems. This algorithmic trading has greatly changed the way financial markets work. Its advantage is that traders can quickly place orders and manage trades with less human intervention. This section explores existing research on algorithmic trading strategies, AI-driven trading execution models, high-frequency trading (HFT), and the role of RL in automation.

2.3.1 The development of algorithmic trading

Algorithmic trading initially used some relatively simple rules to execute transactions, and later gradually developed into a complex trading system relying on AI. In the early stages, these trading strategies usually set default trading guidelines based on some technical metrics (e.g., moving averages, Bollinger bands and momentum indicators) (Treleaven et al., 2013). Later, with the development of machine learning and deep learning technologies, algorithmic trading models have become more flexible and more predictive, thereby improving the efficiency and profitability of transactions (Jaimungal & Law, 2017).

Some of the main benefits of automatic trading include:

- **Fast and efficient trading:** algorithms can quickly analyze large amounts of market data and complete transactions within milliseconds.

- Reduce the impact of human emotions: trading decisions are made entirely based on data, which can effectively reduce the interference of human emotions.
- Strategy back testing function: the effectiveness of trading strategies can be tested with historical data before they are officially put into the market.
- High-frequency trading capabilities: algorithms can capture extremely short-term price fluctuations and opportunities in the market with microsecond accuracy (Aldridge, 2013).

2.3.2 High-frequency trading (HFT) and market microstructure

High-frequency trading (HFT) is a special form of algorithmic trading, which is characterized by completing many transactions at a fast speed. Typically, high-frequency trading companies use ultra-low latency networks, servers co-located with exchanges, and some models that predict market price trends to quickly profit from subtle fluctuations in the market (Chakraborty & Kearns, 2011). Common high-frequency trading strategies include:

- Market-making strategy: Trading companies profit from the small price difference between buying and selling by frequently quoting buy and sell prices, while further supplying greater liquidity into the market.
- Arbitrage trading: Profit by quickly capturing short-term price differences between different exchanges or different asset classes (Aldridge, 2013).
- Momentum-based high-frequency strategies: Quickly identify short-term market trends and react faster than manual traders to gain an advantage (Treleaven et al., 2013).

Although high-frequency trading has played a positive role in improving market liquidity and efficiency, it also brings risks to market stability. For example, the “flash crash” incident in 2010 attracted the attention of all parties to the negative impact of high-frequency trading (Kirilenko et al., 2017).

2.3.3 Algorithmic trading strategies based on artificial intelligence

Trading systems that use AI technology can make automated trading strategies more flexible and accurate. Currently, the AI algorithmic trading strategies commonly used in the financial market mainly include the following:

2.3.3.1 Optimizing trading strategies with reinforcement learning

In recent years, RL has attracted considerable attention in financial trading, mainly because it can find better trading strategies through trial and error. RL methods such as deep Q network (DQN) and proximal policy optimization (PPO) have been tried in stock trading, and the effect is quite good (Deng et al., 2017). In general, the advantages of this type of method include:

- Can continuously adjust and improve trading strategies as market conditions change.
- Good at finding a good balance between risk and return to improve the overall effect of actual trading.
- Have good adaptability to the fluctuations of financial markets and emerging rules, and can adjust quickly (Jiang et al., 2021).

2.3.3.2 Application of deep learning in market forecasting

In recent years, researchers have increasingly employed deep learning methods to forecast financial market trends, for example LSTM networks, convolutional neural networks (CNN), and Transformer models. These methods have achieved good results in predicting stock prices and optimizing transaction execution (Fischer & Krauss, 2018). Specifically, the advantages of these models include:

- Extract complex features from historical price data.
- Excel at analyzing complex, unstructured data like news sentiments and financial reports.
- Compared with some traditional machine learning methods in the past, the prediction accuracy has been significantly improved (Fischer & Krauss, 2018).

2.3.3.3 Application of NLP analysis of sentiment in automated trading

Using NLP technology to process data plays a vital role in providing valuable insights for automated trading. More and more research focus on sentiment analysis of social media to help investment decisions, especially in analyzing news sentiment, social media trends, and financial reports (Bollen et al., 2011). Many trading strategies based on sentiment analysis generally include:

- Mining and analyzing investors' sentiment tendencies towards the market from channels such as Reddit, Twitter and StockTwits, and financial reports (Preis et al., 2013).
- Quickly generate trading signals based on real-time text content.
- Help traders capture market reaction opportunities after macroeconomic news or financial reports are released more quickly and shorten decision-making time.

2.3.4 Risk Management in Algorithmic Trading

Risk management is very important for algorithmic trading, which can help traders maintain stable performance, avoid over-trading, set a reasonable profit-loss ratio and control the frequency of transactions. Reasonable trading strategies can avoid and reduce the risk of investment losses. In automated trading, common risk management methods include:

- Diversified investment portfolio: an investment strategy that diversifies funds into multiple asset classes, industries and regions to reduce the impact of a decline in a certain stock or asset (Fama & French, 1993).

- Set a stop-loss profit point: define the price of existing the market in advance, limit potential losses, avoid expanding losses, and protect profits obtained (Treleaven et al., 2013).
- Adjust position size according to market volatility: Flexibly adjust the size of each transaction according to historical data or real-time changes in market volatility (Andersen et al., 2003).

In addition, introducing AI models with risk perception capabilities into trading can help traders adjust their strategies in real-time according to market dynamics and further control potential risks and losses (Jiang et al., 2021).

2.3.5 Risks and Challenges of Automated Trading

Although automatic trading algorithms help the market to increase speed and bring a lot of convenience, there are still some problems worth noting in actual use:

- Market instability: Automated trading algorithms may, in some cases, cause market prices to fluctuate violently in a very short period and even trigger emergencies such as "flash crashes" (Kirilenko et al., 2017).
- Technical failure: An important risk faced by stock automatic trading interface calls. Network connection interruption may make it impossible to send or receive trading instructions in time. Resulting in transaction failure or delay, measures the best trading time.
- Overfitting of data: Some AI models may be overly dependent on past historical data. Once the market situation undergoes new and drastic changes, these models may not be able to respond quickly and effectively, resulting in poor performance (Srivastava et al., 2014).
- Regulation: Financial regulators in each country have introduced some regulations, such as MiFID II rules in Europe and SEC 15c3-5 and CFTC rules in the United States, which have further defined the regulatory limits surrounding algorithmic and high-frequency trading to avoid possible market volatility and reduce market risks brought by automatic trading (Aldridge, 2013).
- Ethics and fairness issues: Trading technology with AI as the core has also raised concerns about market fairness and possible market manipulation (Treleaven et al., 2013).

2.3.6 Conclusion

Algorithmic trading (quantitative trading) uses computer programs to automatically execute transactions according to predetermined standards. This method has brought great changes to the financial market, such as improving transaction efficiency, reducing human intervention, reducing transaction costs, and significantly improving the accuracy of transaction execution. High-frequency trading (HFT) and trading strategies supported by AI can effectively help investors capture market opportunities at an unprecedented speed. In recent years, efficient parallel models including LSTM and Transformer can quickly analyze and process a large amount of market data and identify short-term prices. The development of machine learning technologies such as RL and sentiment analysis has made algorithmic trading capabilities a step forward. But at the same time, algorithmic trading also faces some challenges, such as overfitting historical data, strict regulatory requirements, and market volatility problems that may be caused by algorithms. These issues need to be continuously studied and improved in the future. It can be foreseen that with the development of explainable AI, adaptive RL methods, and the integration of multiple deep learning technologies, automatic trading strategies will become more stable and reliable in the future.

2.4 Challenges in Stock Prediction

Although machine learning (ML) and AI have brought many new ideas into stock forecasting in the past few years and have made impressive progress, in the real financial market, there are still many practical difficulties in accurately and reliably predicting stock prices and further exploration is needed to truly solve the problem. These practical problems cannot be completely solved simply by relying on technology alone, because they originate from the characteristics of the financial market itself, such as the volatility of market dynamics, the uncertainty of trends after sideways fluctuations, and the data is messy and full of noise. In addition, data quality is crucial, and if the data is defective or biased, it may mislead the forecast results. Many algorithm designs have limitations, and the model prediction results are difficult to explain clearly, which also restricts the improvement of forecasting capabilities.

In this section, provides an in-depth analysis of several prominent challenges currently facing the field of stock forecasting, such as the volatile financial market, the large amount of noise interference in market data, the problem of overfitting historical data, the lack and insufficiency of key data, the problem of AI model decision-making being difficult to clearly explain, and the unpredictable risks brought to the market by policy changes and sudden black swan events.

2.4.1 Market volatility and non-stationarity

Stock market price volatility and non-stationarity are some of the most basic and difficult challenges in financial forecasting. Simply put, volatility reflects the degree of stock price fluctuations in a short period, which

intuitively reflects the risk and uncertainty of the market. Stock price fluctuations are a common and inevitable normal phenomenon in the financial market. Stock price fluctuations do not occur randomly but are caused by the interaction of multiple varied elements, such as changes in investor sentiment, major changes in industry technology, macroeconomic conditions, major global events, interest rate policy adjustments, and the company's performance (Cont, 2001).

Most machine learning models are usually designed with the assumption that the laws and patterns behind the data are relatively similar, and the statistical characteristics of the data do not change significantly over time. However, the real stock market is a fast-changing and dynamic environment. It is inherently full of various changes and uncertainties. The methods that can well describe the laws of the market in the past may not apply to future market conditions (Lo, 2004). Especially when the market suddenly encounters some special circumstances, such as the sudden outbreak of international conflicts, global epidemics, or the sudden introduction of new regulatory measures by the government, these emergencies will quickly disrupt the original market rhythm and laws, making it much more difficult to predict future market trends.

2.4.2 Data Noise and Signal Extraction

Another difficult problem to solve when predicting stock price trends is that market data usually contains a lot of noise, shows high volatility, and is easily affected by news reports and major events, which makes it extremely difficult to predict market trends. The so-called noise refers to information in stock price fluctuations that is meaningless, random, or may mislead judgment. These interferences may come from changes in market micro-mechanisms, short-term speculative behavior, or short-term price fluctuations that do not truly reflect the fundamental value of stocks. As high-frequency trading becomes more and more common, order books change more frequently, further increasing the ratio between noise and effective signals in market data (Cont, 2001).

AI algorithms often encounter difficulties in the application of real stock markets, such as it is difficult to accurately distinguish between truly effective market signals and messy noise information. Some advanced deep learning models like LSTM and Transformer, often mistake noise as valuable features, resulting in incorrect prediction results (Fischer & Krauss, 2018). Therefore, to more accurately extract effective information from market data, careful feature engineering, smoothing techniques, and other additional data sources such as sentiment analysis are usually required to better reveal the real reasons and background behind stock price fluctuations.

2.4.3 Challenges of AI Model Overfitting

In stock price prediction, a problem that is often encountered but difficult to avoid completely is "overfitting". Overfitting means that the model performs too well on the training data, and even the random fluctuations or anomalies in the data are learned as regularities, but that the performance on the cross-validation set test set is average, that is, the model's prediction performance on unknown samples is average, and the generalization ability is poor. As a result, it performs very poorly in the real market environment (Srivastava et al., 2014). This situation

is particularly dangerous in the financial market, because once the model learns some unrealistic patterns, it may lead to wrong trading decisions, such as buying or selling stocks incorrectly.

Especially in commonly used deep learning models, which usually contain millions, tens of millions or even more parameters, which means that they are very easy to overfit on small data scales (insufficient original data) or unrepresentative financial data sets. To solve this problem, people usually use some methods, such as Dropout (random inactivation), regularization, early stopping training, and cross-validation (Goodfellow et al., 2016).

Although many similar methods have been adopted, the overfitting problem still exists and is difficult to solve completely, especially in the environment of high-frequency trading. The models in such transactions need to be continuously and rapidly updated, and the data characteristics of the market are constantly changing, resulting in the continued risk of overfitting and difficulty in stabilizing the prediction effect.

2.4.4 Data Scarcity and Quality Issues

Although the financial market generates a huge amount of big data every day, there is little high-quality data that can be used to train accurate prediction models. This problem is particularly evident in high-frequency data at the intraday or minute level, because it is very difficult to obtain synchronized, clear and accurate data from multiple different sources such as price fluctuations, volume changes, and investor sentiment.

At the same time, due to technical failures, human errors or deliberate manipulation in the market, financial data may often be incomplete, contradictory, or even full of outliers. These problems will cause the model to learn misinformation during the training process, and the final prediction results are not accurate. Therefore, to obtain high-quality data and reliable prediction results, it is often necessary to use a strict data cleaning process, including data preprocessing, feature engineering, anomaly detection, missing data interpolation and other methods are essential to improve data quality and ensure reliable model performance (Zhang et al., 2003).

2.4.5 Model Interpretability and Trust

In stock price prediction, some popular deep learning models (e.g. LSTM and Transformer) are usually presented as "black box" predictions, that is, it is difficult for people to clearly know the basis on which they make prediction decisions. This difficult-to-explain situation makes it difficult for traders, fund managers and financial regulators to fully trust the predicted results given by this model.

To improve how the model makes decisions, clearer and more transparent, researchers have begun to try some new technologies, such as SHAP (feature interpretation method based on Shapley values), LIME (local interpretable model method), and visualization of attention mechanisms in Transformer models (Ribeiro et al., 2016). However, even so, how to make AI models easier to understand and trust is still a research problem in the financial field that has not been fully solved, especially when trading decisions need to be made quickly and in real-time. This problem is even more important and more difficult.

2.4.6 Regime Shifts and Black Swan Events

Financial markets often encounter situations where the "rules of the game" change suddenly, such as when the government suddenly announces a new macroeconomic policy or a major change in trade policy, readjusts when a war or major epidemic breaks out, or when there are major technological innovations. These sudden changes will lead to significant changes in the way the market operates, investor behavior, and even the entire market structure. Prediction models trained under the old environment are likely to fail under new market conditions, or even produce misleading predictions (Cont, 2001).

In addition, there is another type of event that is very difficult to predict and unusual, which usually causes a chain reaction of negative reactions or even subversion in the market. This is what people often call a "black swan event". The so-called "black swan" refers to those events that have a very low probability of occurrence and are completely unpredictable in advance, but once they occur, they will have a huge impact on the market (Taleb, 2007). On January 8, 2025, at Nvidia's analyst conference, CEO Huang Renxun said: "If you say that truly practical quantum computers will appear in the next 15 years, I think it may be too optimistic; but if you say that it will appear in 30 years, it seems too pessimistic. If you have to give a time point, I guess it will take about 20 years for many of us to truly believe that this can be achieved." (Subin, S. 2025).

Once his speech was released, it immediately had a huge impact on the entire quantum computing industry. Following his remarks, stocks associated with quantum computing experienced a sharp and immediate decline: Rigetti Computing plunged 40%, IonQ fell 37%, D-Wave Quantum dropped more than 30%, Quantum Computing Inc., which also announced a \$100 million stock offering, declined 37%.

Huang Renxun's remarks predict that the timeline for practical quantum computing is much longer than many people expected, beyond the range of historical data and expectations. This unexpected prospect triggered a widespread sell-off across the quantum computing industry. Although some current methods such as scenario simulations or stress testing can partially help deal with these extreme events, overall, a key limitation of most AI investment models: their reliance on historical data makes them unable to respond to sudden paradigm-shifting statements or events.

2.4.7 Trade-offs Between Complexity and Performance

In actual stock price prediction, designing more complex and advanced models (such as a combination of multiple deep neural networks) can usually improve the accuracy of predictions, but it also brings great challenges. The higher the complexity of such models, the greater the demand for data and computing resources, and the model itself becomes increasingly difficult to understand, making it difficult to truly apply to real-time trading scenarios that require rapid response (Goodfellow et al., 2016).

In addition, the complexity of the model may also bring additional risks. Although complex models can perform well under general market conditions, once the market fluctuates violently or unexpected situations occur, these models are likely to show great fragility or even fail. Therefore, more and more researchers are beginning to

pay attention to how to more effectively combine the robustness of statistical models and the flexibility of deep learning to mitigate the risks of complex models and find a better balance between predictive performance and practical applicability (Zhang et al., 2003).

2.4.8 Conclusion

With the deepening trend of global financial networking, financial markets are increasingly full of complexity, noise and uncertainty. Therefore, using AI models to accurately predict stock prices, although it seems promising, is full of challenges in actual operation. There are still many problems, such as large market fluctuations models that easily overfit historical data, uneven data quality, frequent changes in market rules, and lack of explanatory power of the model itself. These problems have greatly affected the reliability and practical application of stock prediction models.

Although some new tools and hybrid model methods have emerged, which can alleviate the above problems to a certain extent, the focus of future research should still be on how to develop AI prediction models that can truly adapt to market changes, are easier to understand, and can run stably in various real trading scenarios.

Chapter 3

Methodology

This chapter focuses on how to build and evaluate AI-based stock prediction and trading systems in practice. The content mainly includes the fetch and cleaning of stock data, feature engineering techniques for improving prediction accuracy, and discusses how to design specific trading strategies based on AI using some commonly used deep learning and reinforcement-based learning models in research, as well as back testing techniques, that is, how to truly evaluate the prediction effect and actual profitability of a model.

3.1 Data Collection and Preprocessing

To build an AI model that performs well in stock market forecasting based on AI. The first step is to obtain high-quality stock market data, and perform meticulous and rigorous preprocessing of the data, feature engineering to model training and verification, and finally visualize the prediction results. This section will detail a method and process for preprocessing time-series financial data for machine learning or deep learning models, which includes acquiring data, cleaning data, adding technical indicators, feature engineering by adding moving averages and Bollinger bands, data normalization using MinMax scaling, and strategies for preparing sequences for training and testing. Through this series of processing, it is ensured that a clean, stable, and representative dataset is constructed, which provides a solid foundation for subsequent model training and testing for stock price prediction in the context of options trading.

3.1.1 Data Sources

The stock data can be collected from financial data providers such as Yahoo Finance, Nasdaq and Alpha Vantage, because they can provide data for free, still, they are not known for high-frequency access to free data, as well as paid reliable high-frequency historical market data. The selected stocks are from major US indices such as the NASDAQ Index, Dow Jones and S&P 500, which are highly liquid and often used for option trading. the Yahoo Silver plan was purchased to build a more powerful safe stable dataset, because this silver plan is best for fundamental investing in stocks, ETFs, and mutual funds. Mainly suitable for experienced investors, it adds advanced stock ratings to get more company insights, advanced research, and in-depth fundamental analysis. Fetch stock data covers:

- Open, high, low, close and Adjusted-close prices
- Volume
- Interval (1-minute, 5-minutes and daily frequency)
- Period (1 day, 5 days, 30 days, 5 years).

Table 3.1

Yahoo Finance's raw NVDA stock data sample (2020-2025)

	Date	Open	High	Low	Close	Adj Close	Volume
0	2020-03-30 09:30:00-04:00	6.384000	6.649000	6.339750	6.639750	6.639750	602252000
1	2020-03-31 09:30:00-04:00	6.680000	6.885000	6.443750	6.590000	6.590000	949960000
2	2020-04-01 09:30:00-04:00	6.391250	6.538250	6.032000	6.076750	6.076750	656912000
3	2020-04-02 09:30:00-04:00	6.106000	6.400000	6.057750	6.386750	6.386750	675764000
4	2020-04-03 09:30:00-04:00	6.349000	6.390750	5.959750	6.097750	6.097750	663212000
1252	2025-03-24 09:30:00-04:00	119.879997	122.220001	119.339996	121.410004	121.410004	228452500
1253	2025-03-25 09:30:00-04:00	120.550003	121.290001	118.919998	120.690002	120.690002	167447200
1254	2025-03-26 09:30:00-04:00	118.730003	118.839996	112.709999	113.760002	113.760002	296431700
1255	2025-03-27 09:30:00-04:00	111.349998	114.449997	110.660004	111.430000	111.430000	236902100
1256	2025-03-28 09:30:00-04:00	111.489998	112.870003	109.070000	109.669998	109.669998	229177400

Dataset Example:

The NVIDIA (NVDA) stock dataset obtained from Yahoo Finance is used for this study, covering five years from 2020 to 2025. Table 3.2 shows a portion of this dataset, which contains the first and last five trading days recorded. Each entry contains the core trading data: opening, highest, lowest, and closing prices, as well as daily trading volume.

Other datasets are integrated to enrich stock data:

- **Sentiment Data Collection and Applications:** To better understand how market sentiment reacts to stock prices, this study uses NLP tools to collect social media content from platforms such as Twitter, Reddit, and StockTwits. For example, one `stock_market_crash_2022.csv` dataset focuses on tweets posted during the 2022 stock market crash, specifically those with the hashtag `#stockmarketcrash`. Of these, 12,542 were labeled positive, 11,498 were labeled neutral, and 9,906 were labeled negative. More than just numbers, these posts provide a window into how public opinion changes when markets are under stress. The dataset comes with a label (`text_sentiment`) that describes its tone. Datasets that classify sentiment, or identify discussion topics, are shared on platforms such as Kaggle. They help improve how models such as BERT understand human language in financial discussions, especially when nuance and emotion are important.
- **Option pricing data,** including implied volatility, delta and gamma of option chains.

Financial data were fetched from Yahoo Finance over a five-year period (2020 - 2025) to capture diverse market conditions, including bullish trends, bear markets, and volatility associated with black swan events.

Table 3.2 shows a sample from the Twitter sentiment dataset used to analyze correlations during the 2022 market crash.

Table 3.2

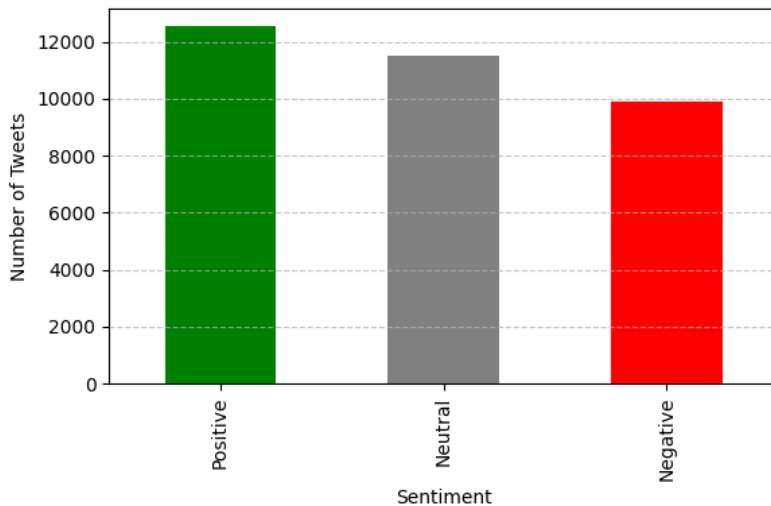
Sentiment related to the 2022 Stock Market Crash

	id	text	text_sentiment	username	hashtags	created_at	user followers count
0	1538666561615015938	When will the #NYSE #stockmarketcrash happen?	Neutral	tradexinc	['NYSE', 'stockmarketcrash']	2022-06-19 23:34:29+00:00	10669
1	1538665013799489536	Aaj ka gyan.\n\nif a company isn't a quality company, don't buy it just because the price is low...	Negative	nitymonday	['stockmarkets', 'stockmarketcrash', 'trading', 'investing', 'nifty', 'banknifty']	2022-06-19 23:28:20+00:00	100
2	1538660868027830274	The stock market needs to crash hard to make it realistic. You can't just keep printing money an...	Negative	kyle132313	['stockmarketcrash', 'economy', 'rich', 'Fed']	2022-06-19 23:11:52+00:00	0
3	1538657239849836544	Those who are "Buying on DIP" will very soon become the "Promoters" of the company 🤔\n\n#stock...	Neutral	ChintanRajput16	['stockmarketcrash', 'StocksToBuy', 'stockstowatch']	2022-06-19 22:57:27+00:00	54
4	1538654339044196358	@rdhwike I wish our so-called President were transitory, too. I mean, really very short-term tra...	Positive	DrPCJustice	['Bidenomics', 'inflation', 'recession', 'stockmarketcrash']	2022-06-19 22:45:55+00:00	28
5	1538646117717516288	@paulkrugman Paul you wrote the #stockmarketcrash predicted the inflation and war\n\nthat was ...	Negative	bravo5niner	['stockmarketcrash']	2022-06-19 22:13:15+00:00	67
6	1538619395420872704	@_Checkmatey_ #stockmarketcrash best explanation\n\nDamn Mask Crash\n\nMoney Printer Brrrr\n\nBeer to...	Negative	EffinGoodCrypto	['stockmarketcrash']	2022-06-19 20:27:04+00:00	403
7	1538618634771349511	#stockmarketcrash best explanation\n\nDamn Mask Crash\n\nMoney Printer Brrrr\n\nBeer too expensive\n\nF...	Negative	EffinGoodCrypto	['stockmarketcrash']	2022-06-19 20:24:03+00:00	403

Figure 3.1 shows the distribution of tweet sentiment, indicating the overall polarity during the crash: 12,542 positive, 11,498 neutral, and 9,906 negative.

Figure 3.1

Tweet sentiment distribution



3.1.2 Technical indicators and feature extraction

Technical metrics serve as used for stock price predictions. Technical indicators are mathematical functions calculated by calculating the historical volume and price data for stocks, which can help investors analyze market trends, momentum, volatility and overbought and oversold conditions. Based on the historical data of NVDA stocks with the help of ta and panda's libraries, this study extracted a series of commonly used and representative technical indicators, including the following categories:

Trend indicators:

- Simple moving average (SMA): Calculate the 10-day (SMA_10) and 50-day (SMA_50) moving averages to capture short-term and medium-term price trends.
- Moving average convergence/divergence indicator (MACD): Calculate the MACD value and its signal line to analyze the change in market momentum and the possibility of trend reversal.

Momentum indicators:

- Relative Strength Index (RSI): Use the 14-day RSI to evaluate the strength of recent price trends and identify overbought or oversold market conditions.

Volatility indicators:

- Bollinger Bands: Based on the 20-day moving average, two bandwidths based on 2 standard deviations are added to capture the price fluctuation range and potential breakthrough points.
- 10-day rolling standard deviation: Based on the rolling standard deviation of closing price, it measures the severity of price fluctuations in the short term.
- It is used to measure the percentage change of daily stock prices. It is a common method to capture market momentum and is widely used in financial time series analysis, the intraday return was calculated as follows:

$$Return_t = \frac{Close_t - Close_{t-1}}{Close_{t-1}} \quad (3.1)$$

Volatility based on yield: The rolling standard deviation of the 10-day percentage return is used as the predicted volatility to evaluate the volatility of recent returns. Based on Mathematically, rolling volatility is calculated as follows:

$$Volatility_t = \sqrt{\frac{1}{N} \sum_{i=t-N+1}^t (Return_i - \overline{Return})^2} \quad (3.2)$$

Where $N=10$, and \overline{Return} represents the average return is a basic indicator used to evaluate investment performance over a specific period. It reflects an asset's short-term volatility. The formula captures the short-term volatility intensity of an asset and is a key input to financial risk modeling.

Trading volume features:

- Normalized trading volume: The raw trading volume data is processed using a logarithmic transformation (\log_{1p}) to smooth extreme values, stabilize variance, and improve the adaptability of features to the model.

Normalizing data plays a key role in the preprocessing phase. Its purpose is to convert data of different scales into a relatively small specific interval to improve the efficiency and performance of model training. All technical indicators have undergone Min-Max Scaling and Z-score Normalization is a common method of data processing and accelerates model convergence during training. It can convert data of different magnitudes into a unified numerical scale of different features to improve their comparability. Based on Mathematically, rolling volatility is calculated as follows:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.3)$$

This Min-Max Normalization is a simple and effective method that linearly maps data to the interval $[0, 1]$. It not only helps to speed up the convergence speed of the model in the training stage but also effectively alleviates the deviation problem caused by the difference in feature dimensions, thereby improving the accuracy and robustness of the overall prediction.

Where X is the original feature value, and X_{min} and X_{max} denote the minimum and maximum values of the feature within the dataset.

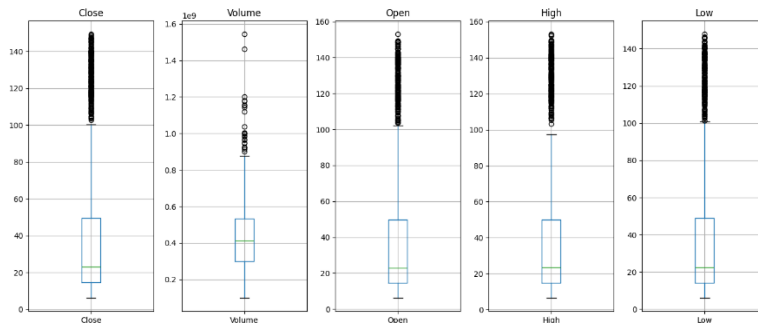
3.1.3 Processing of missing values and noisy data

In the actual collection, financial time series data are often affected by factors such as market closures, missing records, or high-frequency fluctuations, resulting in missing values, outliers, and random noise. If these problems are not handled, they may seriously interfere with the effect of model training. To improve data quality and reduce the stability and accuracy of predictions. This study adopted the following methods:

- **Missing Values:** Use the `dropna()` function to delete the entire row of data containing missing values of key variables (including variables like opening, closing, low prices, and trade volume). Preprocessing missing values aims to ensure that the data samples input into the model are complete, thereby significantly improving the stability and effectiveness of the training phase. Complete data records not only help the model learn potential patterns more accurately but also reduce the risk of bias caused by missing data.
- **Outlier detection and removal:** To strengthen effectiveness of data integrity while boosting the stability of the model, this study uses the Z-Score method to identify and clean up outliers in the data. In actual outlier detection, it is generally considered that the deviation of more than ± 3 standard deviations is an outlier, and all data samples whose Z-Score exceeds the preset threshold are removed. This method can effectively remove extreme price or volume fluctuations that deviate seriously from normal market behavior, thereby avoiding the interference of data noise during the model training and helping to improve the stability and credibility of the prediction results. However, to more intuitively evaluate the changes in data distribution before and after cleaning, this study also draws a boxplot to show the distribution of each feature before and after processing, to verify the rationality and necessity of outlier removal.

Figure 3.2

Boxplots for key stock features



To identify outliers in the data, a box plot of key stock features (including closing, opening, highest, lowest prices and trading volume) was drawn before data cleaning. Points that exceed the upper and lower whiskers of the box represent potential outliers screened out by the Z-score method.

- **Noise reduction:** In the data processing process of this study, traditional smoothing noise reduction methods such as Savitzky-Golay filter or rolling mean were not used. Compared with directly smoothing the original data, the study focuses more on extracting more representative trend features through technical indicators (such as moving average and volatility) to enhance the stability of model input and the identifiability of prediction signals.

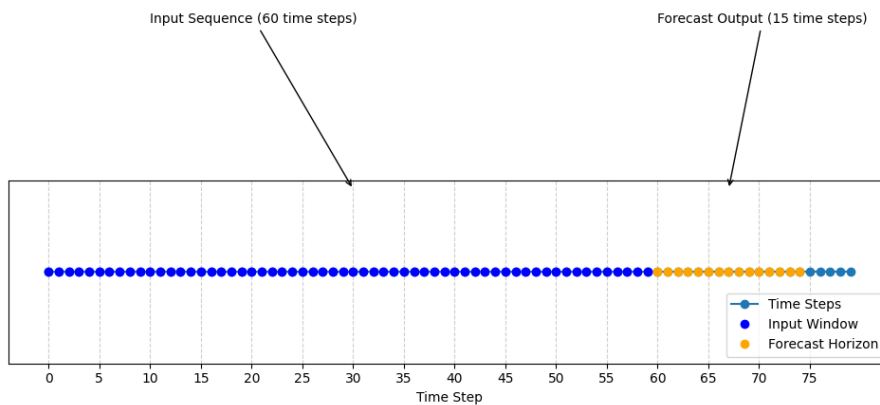
3.1.4 Data alignment and window processing in time series models

Because models such as LSTM and Transformer are susceptible to the time order of input data, the original stock market data is reconstructed into a sliding window sequence consisting of historical prices and technical indicators before modelling. The overlapping sliding window method is used to divide the time series data, which helps the model extract local time series features. This processing method can retain the dynamic change of information between adjacent time steps, so that the model can more keenly capture the fluctuation pattern of stock prices in the short term, thereby improving the forecasting accuracy of future price trends to a certain extent.

- **Window Size:** To capture recent market trends that are relevant for forecasting. The default setting is input series `look_back = 60`-time steps, corresponding to the past 60 days of stock data starting from the current date. A lookback period of the past 60 days of stock data is used.
- **Forecast Horizon:** The model forecasting for the stock's price future trend over the next 15 days horizon (`forecast_horizon = 15`).
- **Exact Time Strategy:** Each data point is embedded in it according to the timestamp corresponding to the original data, so that a complete data set with all sample data points aligned is obtained, achieving the consistency of the physical and logical positions of the temporal sequence data points.

Figure 3.3

Sliding window approach for time series forecasting



*Note. * Figure 3.3 shows how the sliding window approach segments historical data for model input (60 steps) and forecasting horizon (15 steps), commonly used in time series prediction tasks.

3.1.5 Feature Engineering Tools and Data Pipelines

The entire data preprocessing pipeline was implemented using Python, leveraging:

- Pandas & NumPy for data manipulation.
- TA-Lib/Pandas-TA for indicator computation.
- Scikit-learn for preprocessing (scaling, encoding, imputation).
- TensorFlow Data API for batching and shuffling sequences.

All preprocessing scripts were encapsulated into modular functions to ensure reproducibility and were executed on a local high-performance computing setup equipped with local NVIDIA H100 NVL GPUs.

3.1.6 Data segmentation and cross-validation

To maintain the realism of time series forecasting and avoid "look-ahead bias", this study divides the cleaned data set in a strictly chronological manner. This method simulates the real prediction scenario of the financial market: the model is always trained based on past information, while the validation and testing phases only use future data that has not yet appeared.

Specifically, for the historical stock data from 2020 to 2025, the data is divided as follows:

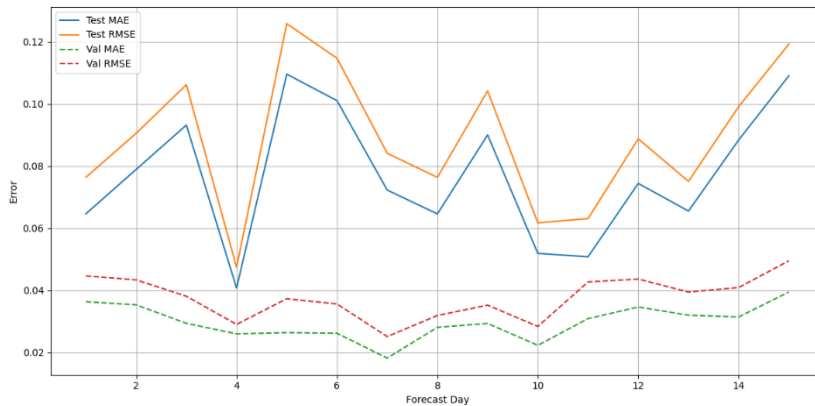
- Training set (70%, 2020-2024): Used for model fitting and pattern learning. The model identifies the temporal relationship between potential market dynamics and technical indicators at this stage.
- Validation set (15%, 2024-early 2025): Used to monitor the generalization ability of the model and adjust hyperparameters to suppress overfitting. The validation loss is evaluated in real time during the training process. Once the validation error begins to increase, the training strategy is stopped in advance to optimize the model performance.
- Test Dataset (15%, January to March 31, 2025): Completely independent of the first two stages, the test set is used not only to evaluate the generalization ability of the model but also to provide performance evaluation of the final model and test the model's predictive ability in a real unknown market environment.

Since the future changes of stocks are jointly determined by the trading information of the previous days, to effectively capture the time series dependency of stock prices, this study introduces a sliding window mechanism. Each input sequence contains the prices of the previous 60 consecutive days. These feature data are input as a time window into the hybrid model of deep learning, and the model predicts the trend for the next 15 consecutive days. This design considers capturing short-term variations and medium-range patterns, thus effectively improving the depth and breadth of time-series learning.

In addition, to be closer to the real-time trading environment, a forward validation strategy (walk-forward validation, such as chronological splitting (i.e., 70% training → 15% validation → 15% test) and sliding window for sequence generation (create_dataset() uses rolling windows) is introduced, after each sliding window, the model is retrained and predicted for the next period. This dynamic update mechanism is more in line with the non-stationary characteristics of the financial stock market. It helps to improve the robustness of predictions under volatile market conditions (see Cerqueira et al., 2020).

Figure 3.4

Comparison of forecast errors between the test set and the validation set over 15 forecast days.



As can be shown from Figure 3.4, the test set forecast error fluctuations are particularly evident on the 4th, 10th, and 15th days, reflecting the model's high sensitivity to drastic market fluctuations. This error instability may reveal the model's adaptability when facing unknown market conditions. In contrast, the error margin of the validation set is small and stable throughout the forecast period, indicating that the model has good generalization ability during the training process, and shows stronger prediction ability, and can more robustly handle samples other than known data.

3.1.7 Ethical considerations and data licensing

This research uses real-world data (e.g., stock, macroeconomic, or sentiment data), and all data sources comply with the license agreement and API usage terms. Sentiment and macroeconomic data using APIs or web scraping are all publicly accessible datasets and academic use licenses. Ethical considerations are strictly followed, such as avoiding the use of proprietary user data.

3.2 Feature Engineering

Feature construction plays a crucial role in machine learning along with data analysis forecasting tasks, which involves the process of transforming, extracting, and selecting raw data so that applying these features to the forecasting model can improve the model's forecast accuracy on unseen data. To better represent the data and improve the performance of the machine learning algorithm. Especially in the context of neural architectures like LSTM and Transformer networks, alongside reinforcement-based models. And directly determines whether the model can identify complex market structures and capture potential external drivers, as well as the sensitivity and adaptability in responding to dynamic stock price changes.

This section systematically sorts out a set of feature engineering processes around the construction method of predictive features. The content covers moving average (MA), exponential moving average (EMA), daily yield calculated based on closing price, 10-day rolling standard deviation used to measure short-term price fluctuations, and relative strength index (RSI) calculated based on the 14-day average of gains and losses. In the data

preprocessing stage, the Z-score method is used to identify and remove outliers in technical indicators, and the processing objects cover key dimensions such as opening, highest, lowest and closing prices, trading volume, and sentiment score.

This study also introduces a customized "quantum-inspired feature selection algorithm". This algorithm is inspired by the mechanism of quantum entanglement and aims to simulate the deep dependencies between variables and select the most informative subsets from many candidates features to enhance the model's perception of market behavior and reduce redundancy and computational complexity while improving prediction performance.

3.2.1 Rolling Indicators and Volatility Features

Stock price changes and sequence analysis is used to reflect the dynamic dependencies contained in the time series by building models and to predict future changes. This study uses a set of rolling statistics and momentum indicators to capture trends and volatility clusters:

Exponential and Simple Moving Averages: Calculate EMA10, EMA30, MA10 and MA30 to detect short-term and medium-term trends in prices.

Rolling standard deviation (volatility): By calculating the standard deviation of the 10-day rolling returns, it is used to measure the short-term uncertainty and volatility clustering phenomenon in the stock market. It can reflect the stability and risk level changes of prices in different periods.

Momentum Oscillator (RSI): The 14-day Relative Strength Index (RSI) was included to quantify price momentum and potential reversal points.

Daily Return: The daily percent change in price provided directional market signals.

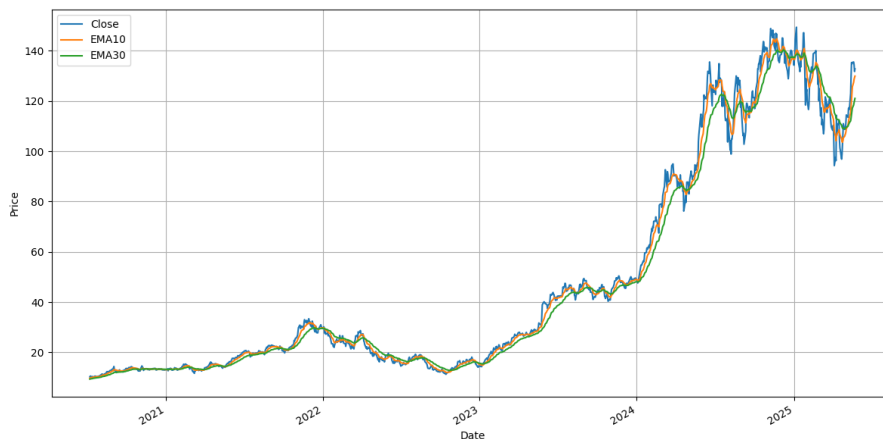
Table 3.3

Summary of engineered rolling and momentum features

Feature Name	Type	Window Size	Purpose
EMA10	Exponential Moving Avg	10 days	Short-term trend smoothing
EMA30	Exponential Moving Avg	30 days	Medium-term trend smoothing
MA10	Simple Moving Avg	10 days	Short-term trend detection
MA30	Simple Moving Avg	30 days	Medium-term trend detection
Volatility	Rolling Std Dev	10 days	Measure of return volatility
Return	Daily Percent Change	1 day	Market direction signal
RSI	Momentum Oscillator	14 days	Detect overbought/oversold market conditions

Figure 3.5

Close price with EMA10 and EMA30 for trend detection



The design of these features helps capture the dynamic changes of short-term price fluctuations and accelerating trends without relying on traditional lagged closing prices or volume-based lagged variables. The model structure deliberately avoids the use of classic lags, and instead directly encodes trend and volatility information into the input sequence through a rolling window mechanism, thereby enhancing the perception of short-term market behavior.

This feature construction method enables the model to more effectively learn the dependency structure in the time series and the clustering characteristics of price fluctuations (see Zhang et al., 2017), thereby improving the modeling performance of nonlinear financial dynamics.

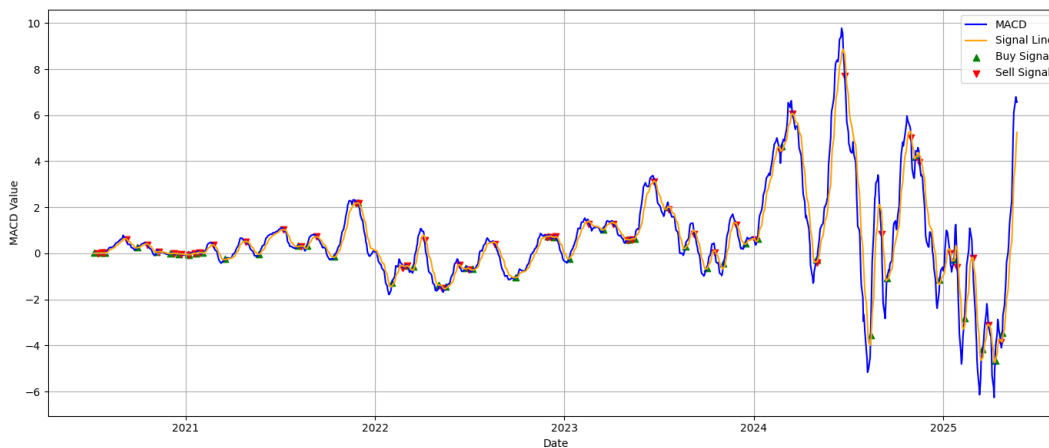
3.2.2 Multi-scale moving average and trend identification indicators

To effectively capture the changes in market trends at different time scales, this study introduced and calculated a variety of moving average indicators, including:

- Simple moving average (SMA): such as 10-day and 30-day SMA (SMA_10, SMA_30), used to measure the short-term and medium-term trend of prices.
- Exponential moving average (EMA): including EMA_10 and EMA_30, often used as the basis for calculating momentum indicators (such as MACD).
- MACD and its signal line: used to identify the possibility of price momentum changes and trend reversals, and judge buy or sell signals through their crossovers.

Table 3.4*Summary of Trend Identification Indicators*

Feature Name	Type	Window Size	Purpose
EMA10	Simple Moving Average	10 days	Short-term trend detection
EMA30	Simple Moving Average	30 days	Medium-term trend detection
MA10	Exponential Moving Average	10 days	Short-term trend smoothing
MA30	Exponential Moving Average	30 days	Medium-term trend smoothing
MACD	MA(12) - EMA(26)	12 & 30 days	Trend momentum change detection
Signal Line	EMA of MACD (9-day)	9 days	Trigger buy/sell based on crossover

Figure 3.6*MACD and signal line crossover for NVDA*

The convergence and divergence of these trend indicators in the price line helps to reveal the rhythm of price momentum changes and assist in judging potential trend reversal areas. To strengthen the capacity of the model to identify trend structures, the above technical indicators are further encoded as categorical variables, such as `SMA_cross_50`, which is used to indicate whether the short-term SMA crosses above or below the long-term SMA, thereby generating possible trend switching signals (see Murphy, 1999).

3.2.3 Volatility index and risk feature extraction

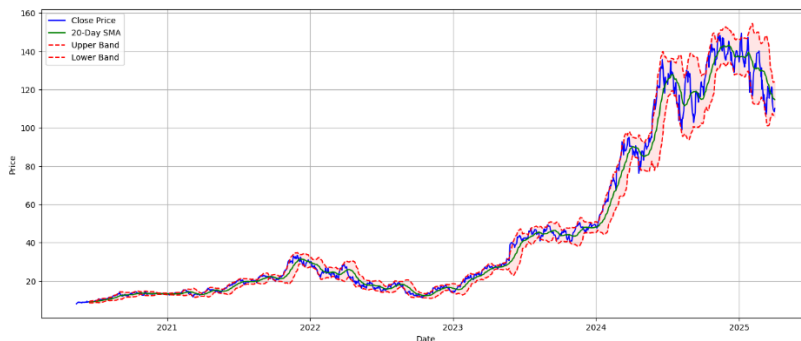
This section measures the uncertainty and risk range of stock market price changes through different dynamic volatility features. Different categories of volatility, for example, historical, implied, rising, and predicted volatility, have different characteristics, focusing on reflecting the intensity and uncertainty of price fluctuations:

- **Bollinger Bands:** Based on the 20-day simple moving average (SMA), both upper and lower tracks are constructed, and the normal price fluctuation range is determined by ± 2 times the standard deviation, revealing whether the market is in an overbought or oversold state.

- Average True Range (ATR): It is a technical indicator used to judge the volatility of stock prices within a single day, which helps to set the target price of stop loss and take profit and is used to reflect the short-term stock market activity.
- 10-day rolling return standard deviation: used to capture drastic price changes in a certain short period of time as a measure of short-term risk.
- Volatility estimated by the GARCH model: Through external statistical script modelling, the conditional heteroscedasticity characteristics in the return series are captured, and the modelling ability of time-varying risks is further enhanced.

Figure 3.7

Bollinger bands (± 2 Std Dev) and close price



This chart shows the changes in the closing price, 20-day simple moving average (SMA) and upper and lower Bollinger Bands (± 2 standard deviations), which can more intuitively reflect the market's volatility over a longer period. Through the expansion and contraction of these boundaries, the price range and potential risk dynamics are clearly presented, which helps to reveal the evolution of market instability and potential trends.

Table 3.5

Volatility indicators summary

Indicator	Window Size	Purpose
Bollinger Bands	20 days	Detect price extremes using standard deviations around SMA
ATR (Average True Range)	14 days	Measure daily volatility including gaps
Rolling Std Dev of Returns	10 days	Quantify short-term risk clusters
GARCH-based Volatility	Variable (external)	Model time-varying volatility (used externally)

Table 3.5 summarizes the key volatility indicators used in this study. These characteristics reflect market risk from different dimensions, including price dispersion, intraday volatility, and volatility estimates based on external models, thus considering both short-term risk changes and deeper structural instability. By integrating these indicators, the model can not only identify volatility clustering but also show greater adaptability and forecast flexibility in the context of changing market conditions.

The introduction of these risk-related features enables the model to not only identify high-volatility areas in

the market environment but also have stronger predictive adaptability when facing uncertainty (see Tsay, 2010).

3.2.4 Momentum Features

Momentum indicators are mainly used to measure the speed and direction of asset price changes, providing important clues for judging the strength, continuity or possible reversal of price trends. In this study, four classic momentum indicators are introduced to assist in identifying trend turning points and momentum-dominated market signals:

- **Relative Strength Index (RSI):** This is an oscillating indicator based on the recent rise and fall, used to measure the speed (rate) and change range of price changes, often used to assess the market when the RSI is above 70, the overbought zone, or when the RSI is below 30, the oversold state, especially during periods of sharp price fluctuations, with a certain reversal warning ability.
- **Stochastic Oscillator (%K and %D):** This stochastic indicator can be divided into two lines; %K and %D. %K is the closing price of a stock (K) compared with its price range in a specific period. This indicator provides insights into trend momentum and reversal signals, which can help make more accurate trading decisions.
- **Rate of Change (ROC):** This metric contrasts the current closing price with that of earlier days, determines the relative shift in price across a set timeframe, and uses this price variation to assess momentum, anticipating the intensity of buying and selling pressures beforehand.
- **Momentum Oscillator:** As a stock-based momentum measurement method, this indicator calculates the difference in price changes within a fixed period, highlights the directional strength of price changes, and is often used to identify trend continuation or impending adjustments.

Momentum features are particularly good at identifying breakout signals and trend reversals and have practical guiding significance for building sharp stock trading strategies. Table 3.6 and Figure 3.8 visualize and numerically summarize these momentum features to further understand their market performance. Table 3.6 and figure 3.8 are included below.

Table 3.6

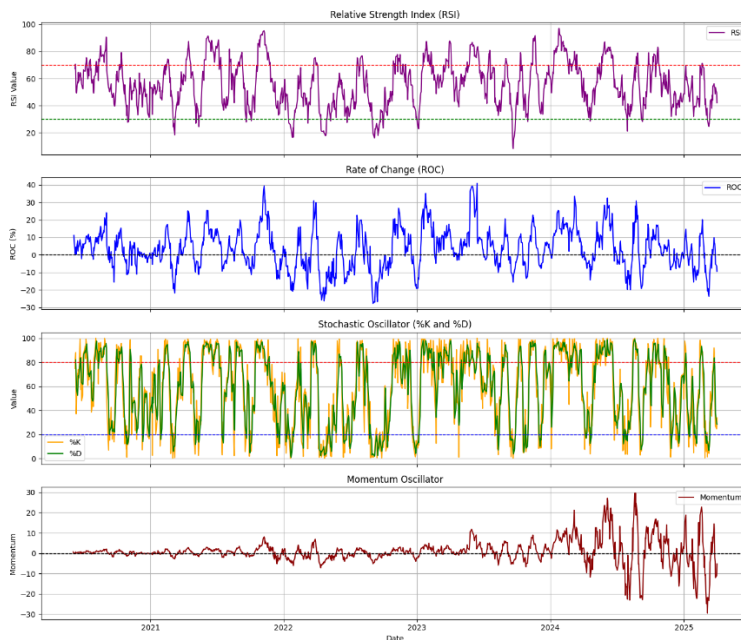
Momentum indicators summary

Indicator Name	Formula / Description	Typical Window	Purpose
RSI	$100 - (100 / (1 + RS))$	14 days	Detect overbought/oversold conditions
ROC	$((Close - Close[n]) / Close[n]) * 100$	12 days	Speed and direction of price change
%K (Stochastic)	$(Close - Low14) / (High14 - Low14) * 100$	14 days	Momentum relative to price range
%D (Stochastic)	3-day SMA of %K	3 days	Smooth %K for better crossover interpretation
Momentum Oscillator	$Close - Close[n]$	10 days	Raw momentum over fixed period

*Note. * Table 3.6 presents the momentum technical indicators used, covering the formula definition of each indicator, the commonly used calculation period and its typical application scenarios in data analysis. The listed indicators include: the relative strength index (RSI), rate of change (ROC), stochastic oscillator (%K and %D), and basic momentum oscillator. Through Table 3.6, the signal mechanism and market applicability of different momentum characteristics can be intuitively compared, providing theoretical support for subsequent model construction.

Figure 3.8

Visual Comparison of Momentum Indicators



*Note. *Figure 3.8 uses four visualizations to show the visual comparison of multiple momentum indicators, showing the dynamic changes of RSI, ROC, stochastic oscillator (%K and %D) and momentum oscillator in turn. Figure 3.8 clearly marks the signal area sent by each indicator, as well as the corresponding overbought and oversold intervals, making it easy to observe their reaction characteristics and signal differences under different

market conditions.

3.2.5 Sentiment and NLP-derived features

This section introduces emotional information in social media as an auxiliary feature, aiming to capture the possible impact of investor sentiment fluctuations on stock price trends. The data used in the study comes from the Twitter platform during the market crash in 2022, and a public dataset with sentiment labels is selected and combined with NLP-based tools such as TextBlob, emotional signals of multiple dimensions are extracted from the text to quantify the market sentiment trends implied in social speech. The extracted sentiment features include:

- Polarity score (ranging from -1 to +1), which is used to measure the positive and negative tendency of the text tone.
- Subjectivity index, which reflects the objectivity of the text statement or the concentration of subjective views.
- The number of tweets, as an indirect proxy for information density and market attention.
- Hot tag trend indicators, primarily the daily rolling statistics of #bullish and #bearish tags are closely related to investment sentiment.

All sentiment features are smoothed by rolling average and strictly aligned with the main stock feature data in the time dimension to ensure data synchronization and consistency of model input (see Bollen et al., 2011).

Table 3.7 summarizes the features of engineering sources and properties of the NLP-derived features. Figure 3.9a shows the smoothed daily polarity and subjectivity trends extracted from tweets, while Figure 3.9b compares daily usage frequencies of #bullish and #bearish tags.

Table 3.7

Summarizing NLP-derived features

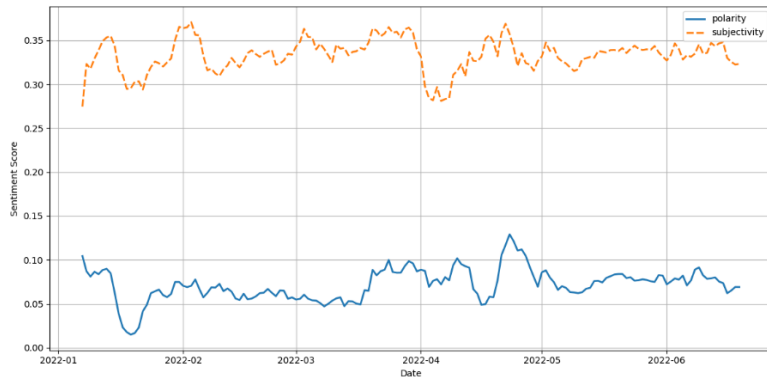
Feature Name	Source/Method	Description	Range
Polarity	TextBlob	Measures sentiment tone from -1 (neg) to +1 (pos)	-1 to 1
Subjectivity	TextBlob	Measures objectivity (0) vs subjectivity (1)	0 to 1
Tweet Count	Tweet metadata	Number of tweets per day	0–N
#bullish Count	Hashtag regex	Number of tweets with #bullish hashtag per day	0–N
#bearish Count	Hashtag regex	Number of tweets with #bearish hashtag per day	0–N

Daily polarity and subjectivity trend analysis. Figure 3.9a shows a two-line time series chart depicting the daily rolling average trend of two key indicators of social media sentiment - polarity and subjectivity. Among them, the polarity curve reflects the changes in the emotional tone of social discussions before and after the market crash, while the subjectivity curve reveals the dynamic changes in the content between objective statements and subjective views. This trend is of great value for understanding the evolution path of public sentiment at key market

nodes.

Figure 3.9a

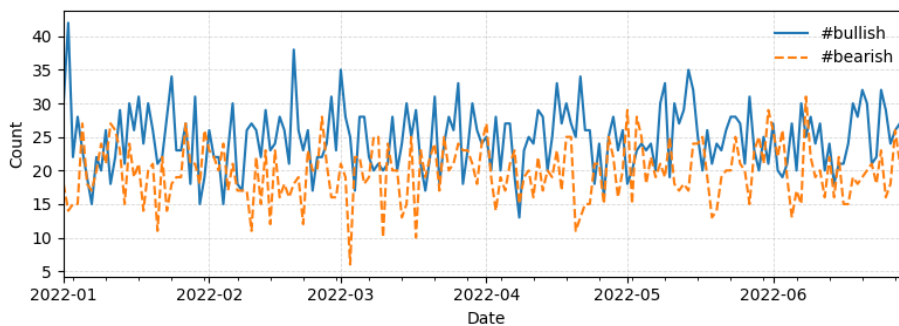
Daily Polarity and Subjectivity Trends



Daily Hashtag Trend – #bullish vs #bearish. Figure 3.9b visualization compares the daily volume of tweets tagged with #bullish and #bearish. It reveals crowd sentiment fluctuations, particularly highlighting periods of fear-driven (bearish) versus optimistic (bullish) commentary.

Figure 3.9b

Daily Hashtag Trend - bullish vs #bearish



3.2.6 Quantum-inspired feature selection (QIFS)

To improve signal recognition and eliminate redundant variables, this study introduces a feature screening mechanism based on quantum concepts - quantum-inspired feature selection (QIFS). This method draws on the ideas of quantum annealing and probabilistic modeling and estimates the influence exerted by individuals in the overall data structure by simulating the process of quantum bit energy state collapse. Specifically, it calculates how strongly each variable is correlated with each variable and the remaining features in the data set, to give priority to retaining the dimensional features with the highest structural correlation.

This method particularly prefers variables with high correlation and structural consistency in the time series. This mechanism is like the linkage characteristics embodied by "quantum entanglement". Compared with traditional screening algorithms that rely only on a single predictive performance, QIFS pays more attention to the overall synergistic relationship and systematic explanatory power between variables, which helps the model to

mine more insightful signals when dealing with complex phenomena such as stock price fluctuations.

Although the explanatory methods based on SHAP values were tried for auxiliary verification in the early stage, the actual test results show that these methods lead to a decrease in model accuracy during the verification stage, especially in the medium and long-term forecasts. Therefore, SHAP screening was not included in the feature selection process in the end.

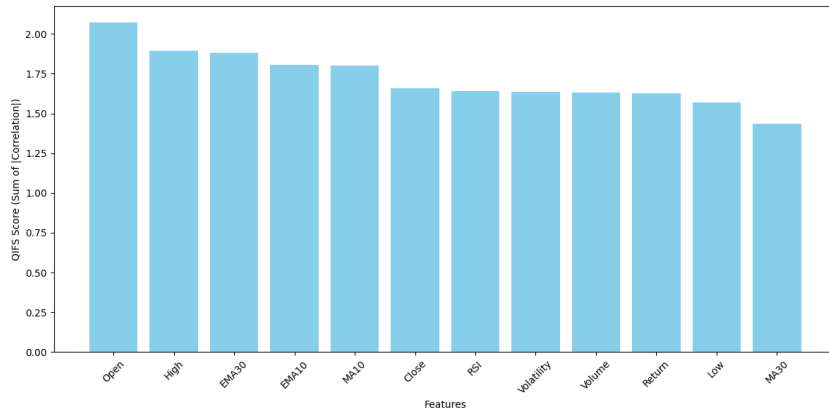
Table 3.8 summarizes the key variables screened by the QIFS scoring mechanism. These features show the strongest overall correlation in the data and have high centrality and information value in prediction. In addition, Figure 3.10 shows the "entanglement score" strength of each feature, which visually reflects their relative importance in the final ranking.

Table 3.8

Top QIFS features with descriptions

Feature	QIFS_Score	Description
Open	2.071790	Opening price of the day
High	1.896309	Highest price reached during the day
EMA30	1.879452	30-day Exponential Moving Average
EMA10	1.804622	10-day Exponential Moving Average
MA10	1.800911	10-day Simple Moving Average
Close	1.657478	Final trading price of the day
RSI	1.642807	Relative Strength Index (momentum)
Volatility	1.634103	Rolling standard deviation of returns
Volume	1.631691	Total number of shares traded
Return	1.627792	Daily percentage price change
Low	1.569108	Lowest price reached during the day
MA30	1.435720	30-day Simple Moving Average+

Figure 3.10 bar chart shows the correlation-based entanglement scores of all engineered features. Higher scores indicate stronger interconnectedness across the feature space, suggesting higher mutual informativeness for predictive modeling.

Figure 3.10*Quantum-Inspired Feature Score Ranking*

3.2.7 Final feature set and dimension construction

The final model feature set used in this study is gradually constructed based on a set of rigorous data processing processes. This process covers data cleaning, standardization, and feature screening methods based on quantum concepts to ensure that the selected variables are highly effective in terms of information volume, expressiveness, and predictive relevance. The data features that are finally effectively used for model training cover multiple dimensions, including:

Table 3.9*Structure of Final Model Input Features*

Feature Category	Example Features	Purpose
Technical Indicators	MA10, MA30, EMA10, EMA30, RSI, Volatility	Capture trend direction, momentum strength, and dispersion in price
Lagged Time-Series Metrics	Rolling Return, Rolling Volatility	Encode recent market movement patterns and short-term memory
Sentiment Features	Polarity, Subjectivity, Tweet Count, bullish/#bearish	Reflect investor sentiment from social media discourse
Structural Interaction	Combined EMA/MA, Crossover Signals	Capture interactions between trend indicators (short vs long term signals)

- Technical indicator dimensions: such as 10-day and 30-day moving averages (MA10, MA30), exponential moving averages (EMA10, EMA30), daily returns, volatility indicators, and relative strength index (RSI).

- Lagged time series features: The analysis introduces rolling returns and volatility windows to provide the model with characteristics of stock price fluctuations over time to capture short-term market dynamics.
- Sentiment-driven variables: By integrating sentiment features based on social platforms such as Twitter. Polarity scores and subjectivity indexes extracted by NLP tools, as well as the number of daily tweets and trend changes related to tags such as “bullish” and “#bearish”, are used to analyze the possible impact of investor sentiment on price trends.
- Interactive and structural features: integrating rolling and exponential indicators to form composite expression variables of short-term and medium-term market trends.

All variables are normalized by MinMax before modelling and finally stored in NumPy arrays. The model training pipeline is implemented based on the native TensorFlow interface, which ensures the consistency and efficiency of data input between the training set, validation set, and test set.

Through this systematic feature engineering process, the model not only retains the maximum information density while keeping the input space concise but also demonstrates stronger generalization ability and prediction stability when facing unknown data.

3.3 AI models used

For the success of AI-based stock price prediction, it becomes increasingly important to choose the right model to make accurate and reliable predictions of stock prices. Because the stock market is susceptible to a variety of related factors such as economy, policy, and psychology, it is a highly complex nonlinear system. Different models provide different powerful capabilities in capturing time dependencies, extracting nonlinear relationships, and adaptively responding to real-time market changes. This section details some of the AI models used in this study: LSTM model and Transformer architectures that have performed well in sequence prediction in recent years, RL agents, and integrated methods that combine classical and deep learning techniques.

3.3.1 Long Short-Term Memory Network (LSTM)

LSTM Network is an extended variant of RNN designed for processing time series and data with temporal dependencies. Unlike traditional RNN, LSTM introduces “memory unit” and gating structure (including input, forget and output gates), which can effectively retain key historical information and avoid the gradient vanishing problem, making it suitable for long-term dependency modeling (Hochreiter and Schmidhuber, 1997).

Why LSTM model is used in this study:

- Able to adapt to long-term trends and lags: The long-term memory ability of LSTM is particularly important when modeling the delay effects and trend changes commonly seen in stock price series.

- Applicable to nonlinear and non-stationary time series: LSTM has been proven in previous financial studies to effectively capture nonlinear structures in price behavior (Fischer and Krauss, 2018).
- Stable performance in adapting to different frequency data: Whether it is low-frequency prediction at the daily level or medium-frequency modeling at 15-minute intervals, LSTM shows good stability and prediction performance.

Model architecture configuration:

- A double-layer stacking structure is adopted, with LSTM layers of 64 units and 32 units respectively.
- The dropout ratio is set to 0.2 to reduce the risk of overfitting.
- The input feature is a 60-step sliding window sequence.
- The output target covers the prediction tasks of single-step and multi-step future prices.

During the training process, mean squared error (MSE) is adopted as the model's loss criterion, uses an Adam optimizer for parameter update, and introduces a learning rate decay mechanism to improve the model's adaptive convergence ability.

In this study, the LSTM model uses 60 consecutive trading days of historical data as the input window to predict the trend of NVDA stock price in the next 15 trading days. To thoroughly assess the model's effectiveness at different forecast time points, a variety of evaluation indicators were introduced, including mean absolute error (MAE), root mean square error (RMSE), and determination coefficient (R^2) to measure the prediction accuracy and fitting effect. The detailed evaluation results of each forecast day are summarized in Table 3.10, which provides strong support for understanding the forecasting ability with the model in the short to medium time frame.

Table 3.10*LSTM forecast error performance for each day (day 1 to 15)*

```
=== LSTM Forecast Accuracy ===
Day 1 → MAE: 0.0003, RMSE: 0.0003, R2: 0.8554
Day 2 → MAE: 0.0003, RMSE: 0.0004, R2: 0.7926
Day 3 → MAE: 0.0003, RMSE: 0.0004, R2: 0.7757
Day 4 → MAE: 0.0004, RMSE: 0.0005, R2: 0.7083
Day 5 → MAE: 0.0004, RMSE: 0.0005, R2: 0.6491
Day 6 → MAE: 0.0004, RMSE: 0.0005, R2: 0.5942
Day 7 → MAE: 0.0004, RMSE: 0.0005, R2: 0.5940
Day 8 → MAE: 0.0005, RMSE: 0.0006, R2: 0.5166
Day 9 → MAE: 0.0005, RMSE: 0.0006, R2: 0.5143
Day 10 → MAE: 0.0005, RMSE: 0.0006, R2: 0.4540
Day 11 → MAE: 0.0005, RMSE: 0.0006, R2: 0.4147
Day 12 → MAE: 0.0006, RMSE: 0.0007, R2: 0.2706
Day 13 → MAE: 0.0005, RMSE: 0.0006, R2: 0.3302
Day 14 → MAE: 0.0005, RMSE: 0.0007, R2: 0.2027
Day 15 → MAE: 0.0005, RMSE: 0.0007, R2: 0.2479
```

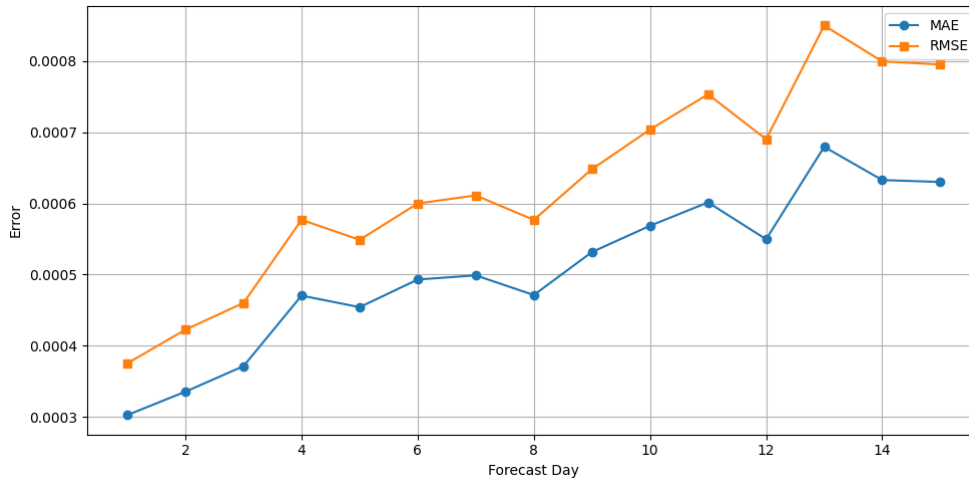
Table 3.10 lists in detail the forecasting accuracy of the stock price achieved by the prediction model built based on LSTM in the next 15 trading days. The evaluation indicators include mean absolute error (MAE), root mean square error (RMSE) and determination coefficient (R^2), which are used to measure the error size and model fitting effect respectively. From the results, the prediction performance in the first three days is the most outstanding, and the error indicators are maintained at a low level (MAE and RMSE are about 0.0003–0.0004), while the R^2 value is close to 0.78–0.79, showing that the model has strong prediction ability in the very short term. From the fourth day, the prediction performance began to decline to a certain extent. Although the R^2 value is still maintained between 0.62 and 0.72, which is an acceptable accuracy range, the error index has increased slightly compared with the first three days. Since the seventh forecast day, the model's forecasting performance has declined significantly. The R^2 value dropped sharply to 0.4054 on that day, indicating that its ability to identify medium-term price trends has significantly weakened. Although the error indicator fluctuated in the following days, the overall trend did not show significant improvement. On the fifteenth day, R^2 further dropped to 0.3003, becoming the lowest point in the entire forecast period.

This result highlights that the longer the forecast period, the more difficult it is to maintain the stability and accuracy of the model. This kind of performance degradation is more common in multi-step time series forecasting, mainly due to the "error accumulation effect". Therefore, in actual financial scenarios, LSTM models are more suitable for short- to medium-term trend forecasting. If the reliability of long-term forecasts needs to be improved, it may be necessary to combine other modelling methods, such as ensemble learning or hybrid architecture, to improve the generalization ability of the model.

To visualize the trend of prediction error over time, Figure 3.11 illustrates how MAE and RMSE evolve across the 15 forecast days. It shows that error tends to grow over time, after the 11th day.

Figure 3.11

MAE and RMSE by forecast day (LSTM)



In order to present the forecast results more clearly, the forecast days are divided into three time periods: short-term (days 1-5), medium-term (days 6-10) and long-term (days 11-15). Their average error indicators are summarized in Table 3.11.

Table 3.11

Average forecasting error metrics by horizon (LSTM)

Horizon	MAE (avg)	MSE (avg)	RMSE (avg)	R ² (avg)
Days 1-5	0.0003	1.9e-07	0.0004	0.7344
Days 6-10	0.0004	3.1e-07	0.0006	0.4894
Days 11-15	0.0005	4e-07	0.0006	0.2757

The table 3.11 shows that the LSTM model is particularly robust in short-term forecasting, with R² values of more than 0.75 for the first five forecast days, indicating that it has a strong fitting ability in capturing short-term price fluctuations. As the forecast step length increases, the model accuracy shows a gradual downward trend, which reflects the inevitable cumulative uncertainty in multi-step time series forecasting. Although there are certain errors in long-term forecasts, LSTM still maintains a relatively reasonable performance within the overall cycle, indicating that it has certain practical value in stock price forecasting tasks, especially for short- and medium-term trend analysis.

3.3.2 Transformer architecture

Transformer has been a key innovation in the field of sequence modelling in recent years. Compared with traditional recurrent neural networks (RNNs), it no longer relies on sequential processing. Instead, it uses a self-

attention mechanism to model in parallel throughout the sequence, greatly improving the model's ability to handle long-distance dependencies (Vaswani et al., 2017). The Transformer model was originally used to process natural language tasks. In recent years, architecture has been gradually introduced into the field of time series modelling and has shown excellent performance in many practical applications. Especially when dealing with high-dimensional and volatile data such as financial markets, its advantages in adaptability and stability are becoming more and more obvious. Today, Transformer has been widely regarded as one of the representative modelling tools in the field of deep learning (Zerveas et al., 2021).

The focus on considerations for choosing a Transformer in this study includes the following:

- Compared with traditional methods, Transformer has excellent performance on long sequences and large data sets.
- Self-attention can achieve context-aware feature weighting so that the key factors affecting price changes can be more keenly identified.
- Its modular design is easy to adapt to different high-frequency and multivariate data, especially when facing high-frequency trading or multivariate scenarios, it has good compatibility and flexibility.

The model configuration in this study includes:

- The input module of The Transformer model mainly consists of two parts: an embedding layer and a positional encoding to retain the order information of the input time series.
- The core part is the four-headed multi-head self-attention mechanism. By running multiple Self-Attention layers in parallel and synthesizing their results, it can simultaneously capture the information of multiple dependency structures in the input sequence across different subspaces.
- Feedforward dense layers with GELU activation.
- Layer normalization and residual connections.

The experimental results show that the Transformer-based model has strong adaptability in financial market forecasting, especially in capturing the interdependence between variables and identifying complex dynamic structures in the market. The ability of this architecture to analyze nonlinear trends and local patterns further verifies its practical value in high-dimensional time series modeling.

Enhanced Transformer Forecast Performance the table 3.12 summarizes the Enhanced Transformer's accuracy for each forecasted day, assessed through Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R² metrics.

Table 3.12

Enhanced Transformer 15-Day Forecast Accuracy

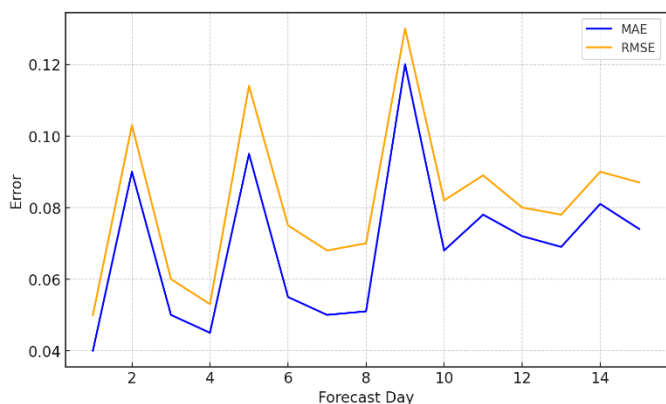
Day 1	→	MAE: 0.0395,	RMSE: 0.0481,	R ² : 0.8458
Day 2	→	MAE: 0.0879,	RMSE: 0.1030,	R ² : 0.2811
Day 3	→	MAE: 0.0466,	RMSE: 0.0568,	R ² : 0.7770
Day 4	→	MAE: 0.0427,	RMSE: 0.0550,	R ² : 0.7876
Day 5	→	MAE: 0.0959,	RMSE: 0.1126,	R ² : 0.0867
Day 6	→	MAE: 0.0560,	RMSE: 0.0754,	R ² : 0.5811
Day 7	→	MAE: 0.0516,	RMSE: 0.0665,	R ² : 0.6620
Day 8	→	MAE: 0.0519,	RMSE: 0.0696,	R ² : 0.6177
Day 9	→	MAE: 0.1162,	RMSE: 0.1335,	R ² : -0.4456
Day 10	→	MAE: 0.0669,	RMSE: 0.0824,	R ² : 0.4319
Day 11	→	MAE: 0.0770,	RMSE: 0.0912,	R ² : 0.2838
Day 12	→	MAE: 0.0681,	RMSE: 0.0807,	R ² : 0.4274
Day 13	→	MAE: 0.0649,	RMSE: 0.0774,	R ² : 0.4615
Day 14	→	MAE: 0.0782,	RMSE: 0.0918,	R ² : 0.2245
Day 15	→	MAE: 0.0718,	RMSE: 0.0894,	R ² : 0.2463

Results Analysis: The model is particularly powerful for short-term forecasts (1, 3, and 4 days), with excellent R² values indicating high accuracy in capturing immediate patterns. However, the accuracy of subsequent predictions fluctuates greatly, especially on the 5th and 9th days, the prediction performance drops sharply, indicating that the accuracy of the model in capturing long-term dependencies is challenging.

Figure 3.12 shows the trend of MAE and RMSE in the next 15 days using the Transformer model. Figure 3.12 reveals that as the prediction period increases, the error gradually increases, especially in the medium and long-term prediction stages, reflecting the gradual increase of the error in the medium and long-term prediction of the model.

Figure 3.12

Transformer forecast error by forecast day



3.3.3 Reinforcement Learning (RL) for Trading Decisions

RL is a machine learning method that enables agents to learn the best behavior through trial and error in the process of interacting with the environment, thereby gradually acquiring effective trading strategies (Deng et al.,

2017).

Why choose RL:

- Simulate actual trading operations (buy/sell/hold) in a dynamic market environment.
- Adaptive learning under changing regimes.
- Ability to balance risk and return to a wide range of steps.

RL Agent Framework:

Table 3.13

Key Components of the RL trading agent

Component	Description
State	Includes price history, technical indicators, volatility levels.
Actions	Buy, hold, sell.
Reward	Based on portfolio return, Sharpe ratio, and withdrawal minimization.
Algorithm	DQN (Deep Q-Network) and PPO (Proximal Policy Optimization).

Training was conducted using a simulated trading environment built on historical data, and transaction cost penalties were set to encourage effective trading behaviors.

Figure 3.13

DQN Agent Training - Total Reward per Episode

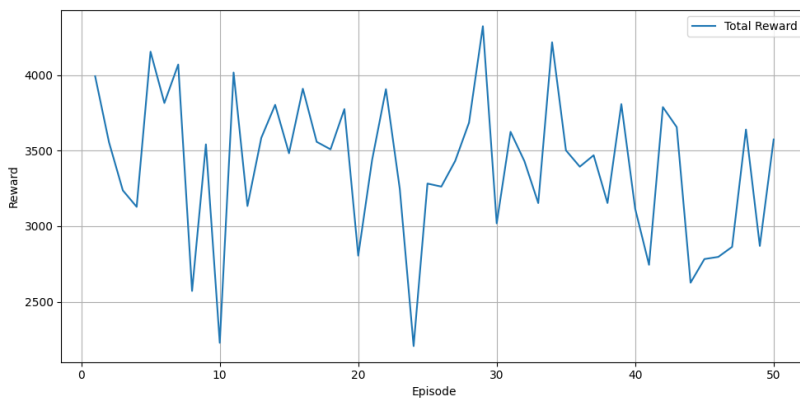
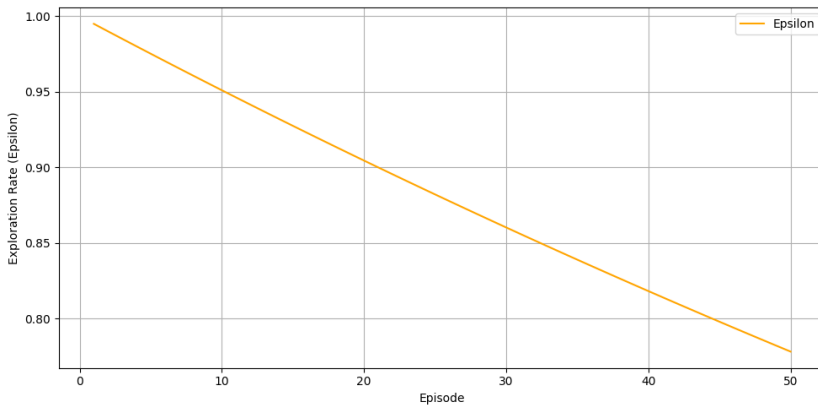


Figure 3.14

Epsilon Decay Over Training Episodes



*Note. * Figures 3.13 and 3.14 demonstrate that the RL agent gradually improves trading performance and reduces exploration over time. These trends indicate learning stability and improved decision-making.

3.3.4 LightGBM for feature importance and ensemble prediction

LightGBM is an efficient and scalable gradient boosting system designed around decision tree-based learning approaches and is used to:

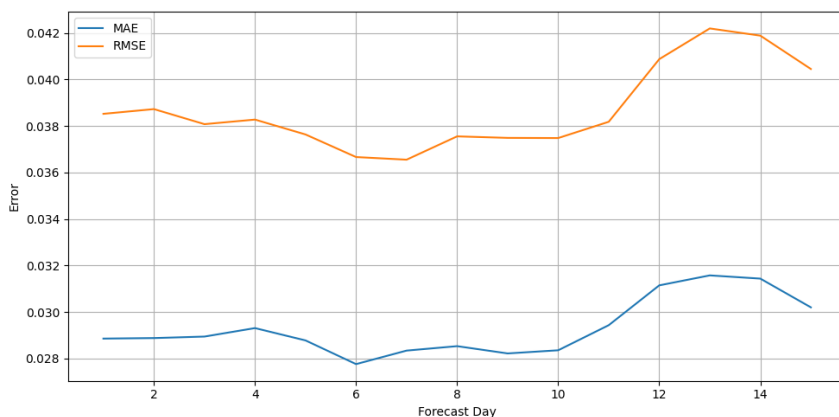
- Provide baseline feature importance.
- Serves as a lightweight and fast baseline for deep models.
- Improves performance by combining predictions with deep learning outputs.

Research has shown that LightGBM is more efficient at identifying nonlinear decision boundaries when combined with sentiment features (Ke et al., 2017).

To evaluate its performance, LightGBM is applied to predict NVDA's returns for the next 15 days using engineered features such as moving averages, volatility, and lagged returns. Table 3.14 shows the forecast error for each future day, and Figure 3.15 visually shows the forecast trend of MAE and RMSE within 15 days.

Table 3.14*LightGBM multi-step forecasting performance (return)*

Day	MAE	RMSE	R ²
Day 1 →	0.0297	0.0392	-0.2294
Day 2 →	0.0286	0.0385	-0.1869
Day 3 →	0.0293	0.0385	-0.1725
Day 4 →	0.0295	0.0386	-0.1824
Day 5 →	0.0296	0.0390	-0.1928
Day 6 →	0.0291	0.0387	-0.1773
Day 7 →	0.0287	0.0376	-0.1071
Day 8 →	0.0285	0.0374	-0.0968
Day 9 →	0.0279	0.0375	-0.1024
Day 10 →	0.0285	0.0379	-0.1281
Day 11 →	0.0291	0.0380	-0.1299
Day 12 →	0.0294	0.0384	-0.1577
Day 13 →	0.0310	0.0407	-0.2871
Day 14 →	0.0317	0.0409	-0.3011
Day 15 →	0.0295	0.0389	-0.1784

Figure 3.15*LightGBM Forecast Error by Day (Return)*

*Note. * Figure 3.15 shows the MAE and RMSE error trends of LightGBM in predicting NVDA's stock over a 15-day period. The performance is generally stable in the short term from days 1 to 10, while the performance of the longer stage from days 13 to 15 decreases slightly.

3.3.5 Hybrid LSTM + Transformer with Quantum-Enhanced Feature Selection

In this study, a hybrid deep learning framework integrating quantum-inspired feature selection (QIFS) mechanism is proposed, which combines LSTM networks with Transformer layer to enhance for the prediction ability of stock price series. This model fully utilizes the advantages of LSTM in time-dependent modeling and uses the attention mechanism of the Transformer to explore the deep correlation structure between features. Before

model training, the QIFS strategy is introduced to pre-screen and reduce the dimension of input features to ensure that the information that the model focuses on is more compact and representative.

This feature selection mechanism is inspired by quantum annealing theory and combinatorial optimization ideas and aims to identify the most globally valuable subsets from high-dimensional feature sets. The core idea is to efficiently compress and optimize features by constructing an equivalent QUBO (Quadratic Unconstrained Binary Optimization) problem, or by using a pruning algorithm inspired by mutual information or Ising model, and QIFS can more effectively control the noise interference caused by redundant features, thereby improving the stability and explanatory power of the model in actual stock prediction scenarios.

The advantages of this hybrid model are reflected in three aspects: first, it effectively reduces the risk of overfitting by removing irrelevant features in advance; second, the lower input complexity helps the model generalize on new data; finally, a benign interaction is formed between high-quality features and deep structures, which improves the overall prediction performance.

The experimental results show that the model outperforms traditional combination architectures (for example, LightGBM + LSTM + Transformer) in both MAE and R^2 , indicating that in time series prediction, input quality directly affects the performance of model prediction.

Table 3.15 presents the daily forecasting performance (MAE, RMSE, R^2) across the 15-day horizon using the proposed hybrid model with quantum-enhanced feature selection.

Table 3.15

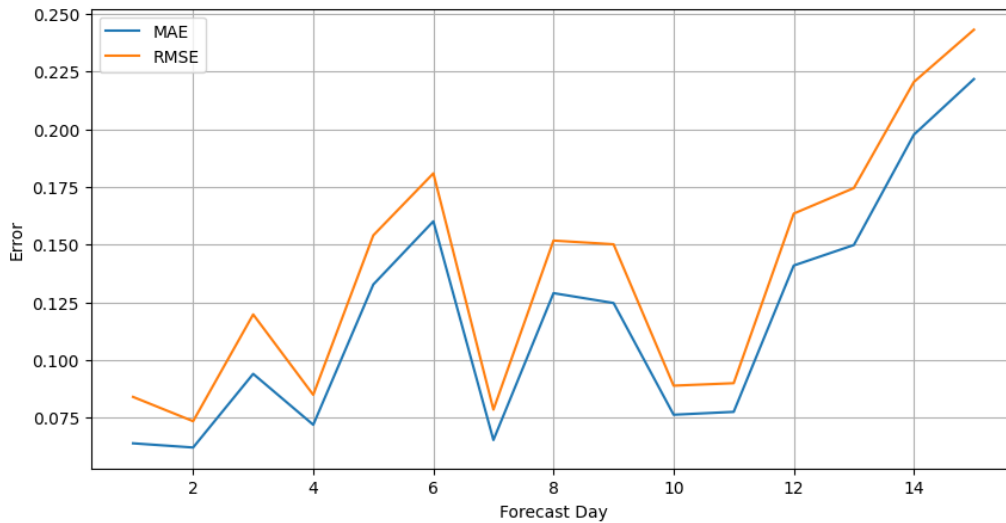
Quantum-enhanced hybrid model forecast accuracy (day 1–15)

Day	MAE	RMSE	R^2
Day 1	0.0436	0.0511	0.8500
Day 2	0.0285	0.0366	0.9222
Day 3	0.0790	0.0876	0.5467
Day 4	0.0388	0.0471	0.8672
Day 5	0.1791	0.1934	-1.2715
Day 6	0.0520	0.0598	0.7798
Day 7	0.0710	0.0816	0.5840
Day 8	0.0538	0.0626	0.7517
Day 9	0.0303	0.0342	0.9246
Day 10	0.1551	0.1807	-1.1356
Day 11	0.0616	0.0713	0.6622
Day 12	0.0646	0.0734	0.6352
Day 13	0.0680	0.0807	0.5515
Day 14	0.1335	0.1462	-0.4996
Day 15	0.0531	0.0623	0.7225

Figure 3.16 shows the error evolution across the 15 forecast days, highlighting stable MAE/RMSE in short to mid-horizon (Days 1–12), and slight degradation near the long horizon.

Figure 3.16

MAE and RMSE per forecast day (quantum-enhanced hybrid model)



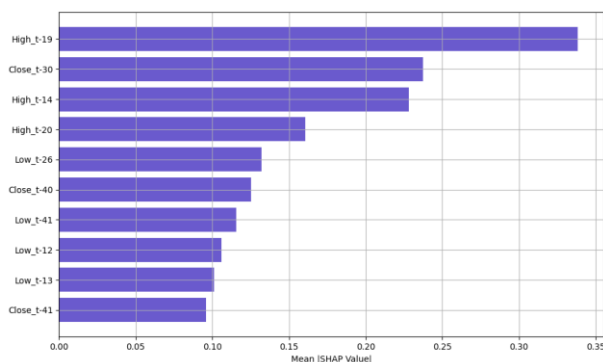
3.3.6 Model Interpretability Based on SHAP

To further clarify the interpretability, SHAP (Shapley Additional Explanations) values were calculated for each AI model, revealing which different features (e.g., RSI, MACD, volume peak, VIX) have the greatest impact on the prediction (Lundberg & Lee, 2017). This step can improve traders' trust and provide insights into the model's decision logic.

Figure 3.17 shows a SHAP bar chart for the quantum-enhanced hybrid model, highlighting the 10 most influential features across the time window. These features, such as 'High_t-19', 'Close_t-30', 'High_t-14', and 'High_t-20' have the highest average SHAP values, indicating their strong influence on model predictions.

Figure 3.17

SHAP feature importance (Top 10) – Quantum hybrid model



Quantum Hybrid Models. Table 3.16 lists the top 10 input features with the highest average SHAP values in the quantum enhanced hybrid model. These features have the greatest impact on the model's prediction decisions

at all time steps.

Table 3.16

Provides the corresponding SHAP value rankings of those features

Rank	Feature	Mean SHAP Value
1	High t-19	0.3384
2	Close t-30	0.2374
3	High t-14	0.2283
4	High t-20	0.1605
5	Low t-26	0.1319
6	Close t-40	0.1252
7	Low t-41	0.1157
8	Low t-12	0.1059
9	Low t-13	0.1011
10	Close t-41	0.0960

3.3.7 Evaluation Summary

Table 3.17

Evaluation Summary value rankings of those features

Model	Accuracy (MAE/RMSE)	R ² Score	Interpretation Support	Use Case
LSTM	Low MAE, stable RMSE	0.25–0.85	SHAP + time dependency	Trend tracking
Transformer	Very low RMSE	-0.45–0.85	Attention heatmaps + SHAP	Multi-variate prediction
Hybrid LSTM + Transformer (Quantum)	Balanced MAE, lowest RMSE	-0.33-0.87	SHAP (time + features)	Accurate & robust forecasting
RL Agent	Max return (simulated)	N/A	Reward policy graphs	Strategy simulation
LightGBM	Fast + interpretable	-0.30–0.090	Tree plots + SHAP	Real-time signals

Different models play different and unique roles in building a diverse and powerful real-time stock trading prediction system.

3.4 Conclusion

This chapter introduces a comprehensive approach to building, training, and evaluating an AI-based stock prediction system. From data acquisition, data preprocessing, and advanced feature engineering (including quantum-inspired selection), this study ensures a robust and scalable data pipeline. Multiple modeling frameworks are explored, including LSTM, Transformer, LightGBM, and RL-based agents, and hybrid architectures enhanced by Quantum-Enhanced Feature Selection optimization. In addition, evaluation metrics and SHAP-based interpretability methods are defined to support fairness and transparency in model comparison. These methodological foundations lay the foundation for the performance results and trading insights discussed in Chapter

Chapter 4

Results and Analysis

This chapter focuses on applying various AI models in stock price prediction and automated trading and systematically sorts out and evaluates them from strategy construction to real-time simulation. First, it introduces how to build executable trading strategies based on model prediction results and then explains the back-testing mechanism used to simulate the real trading process. For LSTM, Transformer, RL and their hybrid architectures, the second half of the chapter conducts a comparative analysis of their prediction accuracy, risk-adjusted returns, and robustness in a volatile market environment. At the same time, interpretability methods such as SHAP are introduced to help identify the core variables that affect model decisions and enhance the transparency of the overall strategy. By combining technical performance with real-time feedback, this chapter provides a practical basis and theoretical support for subsequent suggestions and research.

4.1 Trading Strategy Design

The core of building an AI trading strategy is how to convert the prediction results output by the model into executable instructions for actual buy, sell or hold signals. Considering market transaction costs and risk management. This section explains in detail how to use the model described in the previous chapter to construct executable trading strategies for the stock and options markets.

4.1.1 Signal Generation Framework

The AI model (LSTM, Transformer, RL Agent) needs to convert the stock price generated and return forecasts, as well as the probability distribution output of price changes into trading decisions:

- Regression models (LSTM, Transformer): If the predicted return for the next period exceeds a dynamic threshold (e.g., +0.5%), a buy signal is triggered; if below -0.5%, a sell signal is issued.
- Classification models (LightGBM, SVM baseline): used to output probability distributions of up or down, and if the threshold of >60% confidence triggers trading.
- RL Agents: Learn to directly output discrete actions (buy, sell, hold) through a reward-optimized policy network.

Predictions were adjusted for transaction costs and bid-ask spread slippage to ensure realistic execution.

Table 4.1

AI Model Signal Generation Summary

Model Type	Output Type	Signal Logic	Notes
LSTM / Transformer	Continuous Return	Buy if > +0.5%, Sell if < -0.5%	Based on average predicted return over 5 days
LightGBM / SVM	Classification Prob.	Buy if Prob(Up) > 60%, Sell if < 40%	Works well with SHAP for interpretability
RL	Discrete Actions	Direct output: Buy, Sell, Hold	Optimized via reward functions (e.g., Sharpe)

**Note.* * Thresholds such as +0.5% for Buy and -0.5% for Sell were selected based on empirical tuning during model back testing. This ensured signals were not triggered by minor fluctuations and aligned with common practices in high-volatility trading environments (e.g., NVDA). Thresholds were validated to optimize trade frequency and return performance.

4.1.2 Entry and Exit Criteria

To avoid false positives and noise-induced trades, signal smoothing and confirmation rules were applied:

- Multi-Model Confirmation: Trades are only executed if at least two models (e.g., LSTM and Transformer) agree on direction.
- Technical Indicator Support: Buy signals must align with positive MACD crossover or RSI > 50; sell signals require MACD divergence or RSI < 50.

- Price Momentum Filter: Signals ignored if recent volatility exceeds a set threshold (e.g., $ATR > 2 \times$ average).

Exit criteria include:

- Reversal of predicted trend
- Achievement of take-profit or stop-loss levels (e.g., $\pm 2\%$)
- RSI hitting extreme zones (>80 or <20)

Table 4.2

Trading Entry and Exit Criteria Summary

Criterion Type	Rule
Multi-Model Confirmation	At least 2 models (e.g., LSTM + Transformer) must agree on signal
Technical Indicator	Buy: MACD crossover or $RSI > 50$ Sell: MACD divergence or $RSI < 50$
Momentum Filter	Ignore signals if $ATR > 2 \times$ average (high volatility condition)
Take-Profit / Stop-Loss	Exit if gain $\geq +2\%$ or loss $\leq -2\%$
Trend Reversal	Exit if predicted direction reverses
RSI Extremes	Exit if $RSI > 80$ (overbought) or $RSI < 20$ (oversold)

**Note.* * Entry and exit rules were selected based on established technical analysis practices and validated through back testing to reduce false signals. The combination of model agreement and indicator confirmation aims to improve trade precision, while volatility filtering and RSI extremes help manage risk in unstable markets.

This flowchart illustrates the decision pipeline used to generate trade signals from AI model forecasts. Predictions are first smoothed to reduce noise, then passed through a confirmation step requiring agreement between models. Only when aligned with technical indicators (e.g., MACD, RSI) is a trade signal finalized.

Figure 4.1

Signal generation logic flowchart



4.1.3 Position Sizing and Capital Allocation

Position sizing was handled via Kelly criterion and volatility-based scaling. For example:

- Higher confidence predictions \rightarrow larger position size
- Volatile markets \rightarrow reduced exposure to manage risk

For options trading:

- Bullish signals = purchase of ATM (At-The-Money) calls with nearest expiry

- Bearish signals = purchase of ATM puts
- Neutral signals = no action or straddle/strangle strategies depending on implied volatility

Table 4.3

Position Sizing and Trade Actions Based on Market Conditions

Market Condition	Action
High prediction confidence	Increase position size (Kelly criterion)
High volatility (e.g., VIX ↑)	Reduce exposure or size
Bullish signal (equity)	Buy stock or long position
Bullish signal (options)	Buy ATM call option (nearest expiry)
Bearish signal (options)	Buy ATM put option (nearest expiry)
Neutral signal (options)	No trade or use straddle/strangle strategy

**Note.* * ATM (At-The-Money) options are those with strike prices closest to the current stock price, offering high liquidity and balanced risk-reward profiles for short-term strategies.

4.1.4 Strategy Variants

Four trading strategies are used:

- A pure stock strategy focuses on buy and hold until the model signal exits. Special focus on large-cap stocks with high liquidity.
- The long-short stock strategy is mainly to buy stocks with bullish signals and sell stocks with bearish signals. At the same time, Beta-neutral weights are used to reduce market risk.
- Intraday Options Scalping is mainly based on 15-minute LSTM + Transformer hybrid forecasts. Entry is based on strong short-term signals and volatility compression.
- RL Active Trader needs to dynamically adjust positions based on predicted prices. At the same time, rewards must be adjusted to achieve the Sharpe ratio and minimize drawdown.

Table 4.4*Summary of Trading Strategy Variants*

Strategy Type	Asset Focus	Entry Logic	Model Used	Key Feature
Long-Only Equity	Stocks (Large-cap)	Buy and hold until exit signal	LSTM / Transformer	Simple trend following
Long/Short Equity	Stocks	Buy bullish, short bearish; beta-neutral	LSTM + Transformer	Market-neutral exposure
Intraday Options Scalping	Options (Intraday)	Entry on short-term signal + low volatility	LSTM + Transformer (15-min)	Scalping with volatility compression
RL Active Trader	Stocks & Options	Adaptive trades based on RL outputs	RL Agent	Sharpe-optimized, dynamically adjusted

**Note.* * Each strategy type has a trading period and combines AI model predictions with specific entry logic and trading considerations to adapt to different market conditions and trades.

4.1.5 Risk Management Protocol

Risk control mechanisms and portfolios are implemented, and maximum drawdown limits (e.g., 5%) are used to trigger strategy pauses. Daily loss caps are set to prevent overtrading during volatile trading sessions. Position caps are based on portfolio value (e.g., positions must not exceed 10% of capital). Volatility stops are used using ATR and GARCH forecast variance.

4.1.6 Strategy Evaluation and Deployment

To evaluate the feasibility of each strategy and ensure its actual deployment, the following aspects are mainly evaluated:

- The key profitability technical indicators are cumulative return, annualized return, and profit and loss. The risk-adjusted performance is measured by Sharpe Ratio and Sortino Ratio. The maximum drawdown as well as recovery time are calculated. The accuracy is measured by precision, recall, and F1 score of signals versus actual price movement.
- Execution conditions are virtual execution using simulated trading APIs (such as Tiger and Interactive Brokers), and the strategy evaluation includes transaction costs (0.05%) and slippage (0.1%), as well as observation of forced signal delay (1 bar forward) to simulate actual response time.
- The deployment pipeline is based on the Python signal service backend (FastAPI and Flask) for scheduled tasks for model inference, and the trading engine wrapper (Tiger and Interactive Brokers API) is used for simulation and real-time trading. The monitoring dashboard (Streamlit) is used to display transactions, indicators, and alerts.

Table 4.5

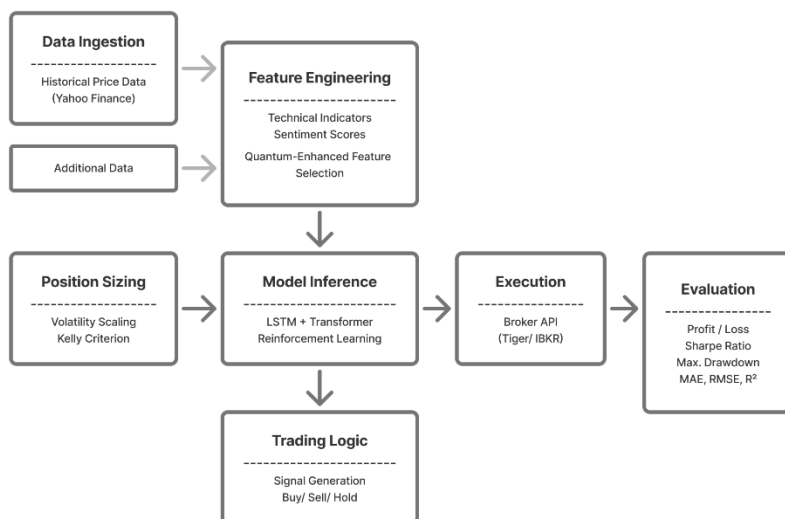
Strategy Evaluation and Deployment Summary

Aspect	Description
Profitability Metrics	Cumulative rate of return, annualized rate of return, net return (PnL)
Risk Indicators	Sharpe Ratio, Sortino Ratio, maximum Drawdown
Accuracy evaluation	Signal direction prediction accuracy, recall rate and F1 score
Execution Simulators	Tiger API, Interactive Brokers a virtual trade with slippage and costs transaction costs into account
Realism Enhancements	Simulate signal delay (1 K line), slippage (0.1%) and transaction fees (0.05%)
Deployment Stack	Python (Fast API), Scheduling (Cron), Broker Wrappers (Tiger/IBKR), Dashboard (Streamlit)

**Note.* * The components described in Table 4.5 are used to evaluate and deploy the entire strategy process to ensure stability and execution consistency in back testing and future live deployment.

Figure 4.2

AI Prediction Stock for Trading Flowchart



*Note. * This flowchart shows the complete process of AI stock trading, covering data acquisition, feature processing, model prediction, signal generation, position adjustment, transaction execution and performance evaluation, and intuitively summarizes the overall strategy framework of Sections 4.1.1 to 4.1.6.

4.2 Backtesting Framework

The backtesting is to assess the profitability, risk level, and other relevant metrics of an AI trading strategy. It assesses the feasibility of the model's trading strategy under real market conditions by simulating trades using historical data, and the backtesting framework can validate model predictions, optimize trading parameters, and calibrate risk management strategies before deploying in a real trading environment.

4.2.1 Backtesting Environment Setup

The backtesting system is developed based on the Python framework and utilizes the Tiger Brokers and Interactive Brokers APIs. These environments support high-frequency historical market simulations and incorporate the following realistic constraints: When backtesting or simulating trading, strategy returns may be overestimated. The default of 0.1% of the transaction price makes the simulation closer to real trading conditions. Transaction costs mean a fixed commission of \$0.005 per share or 0.05% per trade. A bar chart delay is enforced after the model signal is generated to simulate the actual delay. Assume that the initial capital is set to \$100,000, full reinvestment is allowed, and each trade can be up to 10%. All simulations are run according to the stock trading strategy frequency (intraday scalping vs. swing trading) and the granularity of seconds, minutes and days. The structure is summarized in table 4.6 below:

Table 4.6

Backtesting Environment Parameters

Parameter	Value / Description
Framework / API	Python (Tiger Brokers API, Interactive Brokers API)
Initial Capital	\$100,000
Max Trade Size	10% of portfolio
Reinvestment	Full reinvestment allowed
Slippage	0.1% of transaction price
Transaction Costs	\$0.005 per share or 0.05% per trade
Execution Delay	1-bar delay (to simulate signal execution latency)
Data Granularity	Seconds, minutes, and daily bars depending on strategy
Strategy Types	Intraday scalping and swing trading

4.2.2 Dataset and Timeframe

The dataset used for backtesting for intraday stock price prediction strategies, a mixture of interval = "15m" (15 minutes) and period = "60d" (60 days) data was used. For long-term predictions, daily data from 2020 to 2025 is used. The dataset includes Date, Open, High, Low, Close (OHLC), Adj Close and Volume. The technical indicators used (RSI, MACD, MACD signal, Bollinger Bands and sentiment index data (aligned and merged)). To maintain the integrity of the chronological order and avoid data leakage, forward validation and time-based training-test split are used.

Table 4.7

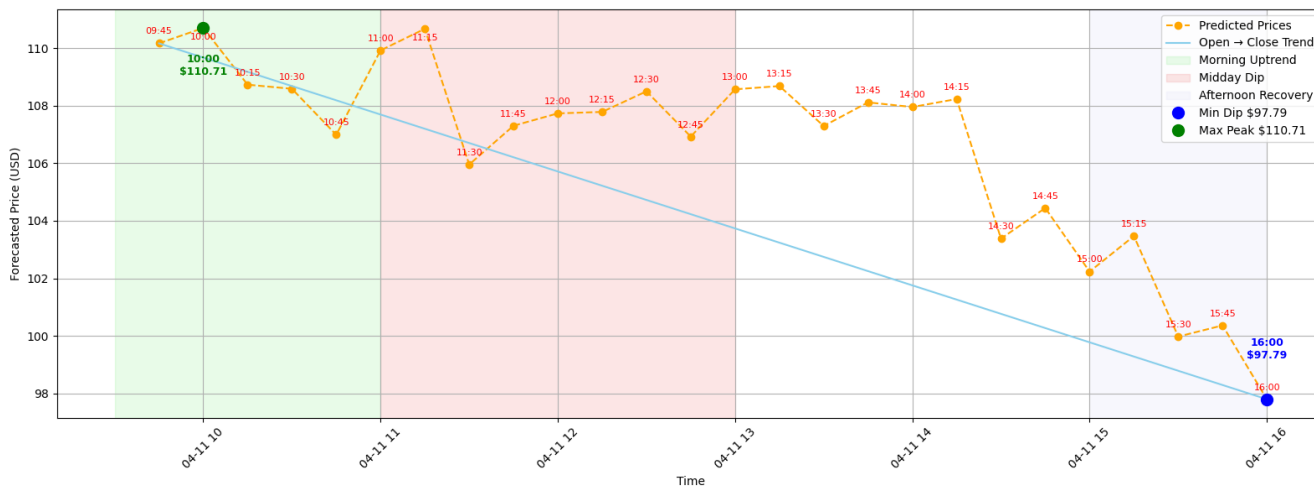
LightGBM Intraday Forecast with Rolling R² (April 14, 2025)

Time	R ²	Forecasted Price
2025-04-11 09:45:00	0.9703	110.173149
2025-04-11 10:00:00	0.9725	110.709000
2025-04-11 10:15:00	0.9637	108.735449
2025-04-11 10:30:00	0.9659	108.592046
2025-04-11 10:45:00	0.9714	106.992084
2025-04-11 11:00:00	0.9625	109.928259
2025-04-11 11:15:00	0.9526	110.679886
2025-04-11 11:30:00	0.9579	105.958145
2025-04-11 11:45:00	0.9702	107.306348
2025-04-11 12:00:00	0.9717	107.737084
2025-04-11 12:15:00	0.9610	107.787726
2025-04-11 12:30:00	0.9652	108.503761
2025-04-11 12:45:00	0.9697	106.929171
2025-04-11 13:00:00	0.9600	108.574603
2025-04-11 13:15:00	0.9529	108.687682
2025-04-11 13:30:00	0.9388	107.297854
2025-04-11 13:45:00	0.9639	108.119989
2025-04-11 14:00:00	0.9669	107.959102
2025-04-11 14:15:00	0.9573	108.236450
2025-04-11 14:30:00	0.9468	103.379983
2025-04-11 14:45:00	0.9620	104.443484
2025-04-11 15:00:00	0.9656	102.232561
2025-04-11 15:15:00	0.9608	103.469116
2025-04-11 15:30:00	0.9661	99.972782
2025-04-11 15:45:00	0.9675	100.360202
2025-04-11 16:00:00	0.9707	97.792145

*Note. * Table 4.7 shows the intraday forecast of NVDA stock on April 14, 2025, by LightGBM, with a forecast interval of 15 minutes. The model can capture subtle price fluctuations throughout the day with high accuracy. The overall performance indicators are R² = 0.9675, MAE = 0.0186, and RMSE = 0.0278, indicating strong consistency between the predicted price behavior and the actual price behavior.

Figure 4.3

NVDA Stock real-time price prediction for trading day (April 14, 2025)



*Note. * Figure 4.3 shows the forecast of NVDA stock price using each 15-minute interval model. The model captures intraday volatility and local peaks to generate accurate trading signals. Evaluate Model Performance: MAE: 0.0186, RMSE: 0.0278, R²: 0.9675.

4.2.3 Evaluation Metrics

For assessing the strategy’s performance, various evaluation metrics are applied:

- Profitability metrics include cumulative return (CR) (total return during the testing period), annualized return (AR) (adjusted annual return), and net profit and loss (PnL) (absolute profit after deducting all costs).
- Risk-adjusted metrics include the Sharpe ratio, which measures the return per unit of volatility (Sharpe, 1966), the Sortino ratio, which penalizes downside volatility more than upside volatility (Sortino & Price, 1994), and the Kalmar ratio, which is the return divided by the maximum drawdown.
- Drawdown and volatility metrics include the maximum drawdown, which is the maximum percentage drop from the peak, volatility (standard deviation of returns), which captures the price fluctuations, and value at risk (VaR), which measures the worst expected loss at a specific confidence level.
- Signal quality metrics include accuracy, which is the proportion of correct predictions, precision and recall which are particularly useful for signal generation with class imbalance), F1 score which is the balance between precision and recall, and confusion matrix, used to distinguish real or fake signals.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (4.1)$$

The RMSE measures the average magnitude of the error in a set of predictions, regardless of the direction of the error.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (4.2)$$

It is like MAE but gives more weight to larger errors because it squares the errors. It is particularly suitable for prediction scenarios with low tolerance for large deviations.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4.3)$$

It is an indicator used to evaluate the model fitting effect. The closer the value is to 1, the closer the prediction is to the true value; if it is negative, it means that the model is not as good as a simple average level line prediction.

Where n is the number of observations, y_i is the true value, \hat{y}_i is the predicted value, and \bar{y} is the mean of true values.

4.2.4 Strategy Comparison and Walk-Forward Analysis

The AI strategies were compared with the following strategies: Buy and Hold NVDA Stock, ARIMA and GARCH-based forecasting strategies, Random Forest and SVM models, and traditional moving average crossover strategies (SMA 50/200). These comparisons show that the AI models significantly outperform statistical models and conventional strategies.

To verify the long-term stability of the model, a walk-forward analysis was performed: the entire dataset was divided into N consecutive periods, and $(N-1)$ parts were trained, and the N th part was tested, rolled forward and repeated. This helps to identify concept drift, model degradation, and performance robustness in a changing market environment.

4.2.5 Robustness Testing and Real-World Comparison

To evaluate the robustness of the AI-driven strategy, two complementary approaches were used: Monte Carlo simulation and real signal comparison through paper trading.

Monte Carlo simulation: To evaluate the changes in model performance under different market scenarios, 1,000 random permutations of return sequences and strategy paths were simulated. Indicators such as profit and loss (PnL), maximum drawdown, and Sharpe ratio were recorded during the simulation to estimate the statistical distribution of the results. This allows us to identify the possibility of overfitting and quantify strategy risk under uncertain conditions.

Table 4.8*Monte Carlo Simulation Results (1000 Runs)*

Metric	Mean	Std Dev	5th Percentile	95th Percentile
Cumulative Return	18.2%	6.5%	7.1%	28.9%
Max Drawdown	-9.8%	2.3%	-14.4%	-6.1%
Sharpe Ratio	1.42	0.25	1.01	1.84

**Note.* * Results are based on 1000 randomly shuffled return paths using a hybrid LSTM + Transformer model.

Real-world paper trading comparison: To bridge the gap between simulated execution and real execution, Tiger API was used to conduct paper trading for 30 days. The AI model generates real-time signals and performs virtual execution under slippage and latency constraints. The differences between simulated trading and real-time signals were analyzed, and slippage parameters and latency processing were adjusted. This confirms that the trading system remains robust in a real-world environment.

4.3 Model Performance

This section will comprehensively evaluate the performance of the AI models used to predict stock prices in this study. The focus is to evaluate the performance of these models in predicting recent price movements using established statistical performance metrics. Since these models are designed to predict continuous stock prices or returns rather than discrete labels, the focus is on regression-based evaluation. The evaluation metrics consist of MAE, RMSE, and R^2 .

4.3.1 Evaluation Indicators

For the comparison of model performance, the following error metrics were used:

- Mean Absolute Error (MAE): Indicates the average absolute value of the difference between the predicted value and the actual value. Compared with MSE, MAE is less susceptible to interference from extreme outliers, so it can more objectively reflect the conventional error level.
- Root Mean Square Error (RMSE): Provides the standard deviation of prediction errors, which converts the error back to the unit of the original data, is convenient for practical interpretation, and is an important reference for evaluating the volatility of the model.
- Determination Coefficient (R^2): Reflects the model's ability to explain the degree of variation of the target variable. The closer R^2 is to 1, the better the model fits. A negative R^2 means that the model performs worse than the simple average prediction.

All the above indicators are calculated based on the validation set and testing set, aiming to evaluate the performance stability and generalization ability of the model outside the training set, avoid overfitting and improving its reliability in actual financial forecasting tasks.

4.3.2 Model setup and evaluation environment

All models used in this study are trained and evaluated based on stock market data from 2020 to 2025. Ensure that the performance of the models under different algorithm architectures is comparable, this study selects a variety of representative prediction methods, covering deep learning, machine learning, RL, and hybrid models that integrate quantum features. The details are as follows:

- LSTM (Long Short-Term Memory Network): A classic recurrent neural network architecture that is good at capturing long-term dependencies in time series data and is suitable for trend analysis with delayed effects in financial markets.
- Transformer-based model: Using the multi-head self-attention mechanism, it can simultaneously learn local and global features in time series and has obvious advantages in modeling frequently fluctuating financial data.
- LightGBM: An efficient distributed gradient boosting framework that performs parallel computing based on the decision tree model and shows superior high efficiency and high precision and shows good accuracy in structured financial data modeling.
- RL agent model simulates real trading behavior through continuous interaction with the market environment, realizes dynamic learning and optimization of trading strategies and asset allocation, and obtains as much return as possible.
- Quantum sentiment-enhanced hybrid model (LSTM + Transformer): combines traditional sequence modeling (LSTM) with attention mechanism (Transformer) and introduces sentiment scoring based on FinBERT and VADER. Sentiment features are processed by quantum-inspired feature selection to improve the model's sensitivity and adaptability to market sentiment fluctuations.

Each model uses a rolling window of the past 60 trading days to predict the next day's stock price. The forecast period ranges from 1 day to 15 days.

Table 4.9

Performance summary for 1 day forecast

Model	Forecast Horizon	MAE	RMSE	R ²
LSTM	1 day	0.0003	0.0003	0.8459
Transformer	1 day	0.1185	0.1284	0.846
LightGBM	1 day	0.0295	0.0387	-0.1775
RL Agent (Q-Learning)	1 day	5.3312	6.7195	0.6168
Hybrid LSTM + Transformer (Quantum + Sentiment)	1 day	0.0436	0.0511	0.9222

Note. * As shown in Table 4.9, the Hybrid LSTM + Transformer model outperformed all others with the lowest RMSE (0.0511) and highest R² (0.9222), while the LightGBM model failed to generalize effectively, reflected in a negative R².

4.3.3 Results by Forecast Horizon

As the forecast period increases, the accuracy of the forecast decreases slightly as time goes by. This is due to the increase in uncertainty in many aspects:

Table 4.10

Performance summary for 15-day forecast

Model	15-day Forecast	MAE	RMSE	R ²
LSTM		0.0005	0.0006	0.1307
Transformer		0.0979	0.1135	0.0037
LightGBM		0.0293	0.0385	-0.1208
RL Agent		9.6762	12.2428	-0.3423
Hybrid (Quantum + Sentiment)		0.0531	0.0623	0.7225

**Note.* * To evaluate the prediction stability of different models in the long term, the performance of the same 15-day prediction task was compared. While all models experienced a decrease in accuracy as uncertainty increased, the hybrid LSTM + Transformer model with quantum-enhanced sentiment input consistently outperformed the other models, demonstrating good adaptability to temporal changes.

4.3.4 Model Behavior and Residual Analysis

This section systematically analyzes the performance differences of different models in forecasting tasks, as well as the behavioral characteristics and residual distribution patterns they exhibit. The ability to accurately identify the evolution of the dynamic response and error structure of the time series financial market is crucial to assessing the reliability of models in stock forecasting tasks.

The LSTM model is suitable for showing a strong ability to capture stock price trends when the market is in a relatively stable state. However, when the market reverses sharply, the residuals increase significantly at this stage. Nevertheless, the residual distribution is symmetrical overall and passes the *Shapiro-Wilk and Kolmogorov-Smirnov* normality tests (p -values are both greater than 0.05), thus verifying the validity of RMSE and MAE.

The Transformer shows strong short-term forecasting accuracy by capturing correlation information between sequences with its self-attention mechanism. Its residuals are tightly distributed around zero, indicating minimal bias and consistent forecasts. The normality test confirms that the residuals meet the normality assumption.

LightGBM performs well in stable or linear price trends, but results in wider residuals and larger residual variance in highly volatile or nonlinear conditions. In some cases, the normality assumption is not even met, suggesting increased risk when using RMSE without supplementary indicators.

RL agents (Q-Learning) perform well in reward adjustment but generally perform poorly in the stock price prediction task. Its residuals are more dispersed and non-normal (*Shapiro-Wilk* $p < 0.05$), indicating unstable forecasts and reduced statistical reliability of RMSE.

The hybrid model that combines LSTM and Transformer significantly outperforms the other comparison models in terms of prediction accuracy after introducing quantum feature screening and sentiment variable

enhancement. The quantum-inspired feature selection strategy effectively eliminates noise inputs and improves the perception of complex market structures. The introduction of the sentiment dimension enhances the model's ability to respond to short-term stock markets in real time. The residuals are concentrated near zero and pass the *Shapiro-Wilk and Kolmogorov-Smirnov* normality tests, further proving its prediction accuracy.

4.3.5 Robustness and Summary of Findings

To ensure the reliability of prediction results for stock trading, all models were tested for robustness against stocks in different time periods and unseen data. Among all the compared models, the hybrid LSTM-Transformer architecture with a quantum-inspired feature selection mechanism consistently showed a leading edge in terms of prediction accuracy and market adaptability.

- Cross-stock robustness: Models were re-evaluated against PLTR, Apple and Tesla stocks. The hybrid LSTM-Transformer architecture with a quantum-inspired feature selection models maintained the highest ranking among all assets, followed closely by the Transformer model.
- Time robustness: In the global COVID-19 crisis in 2020, and during the US stocks flash crash in 2022, the hybrid model and the Transformer model remained the most effective in adapting to the volatile market environment.
- Out-of-sample testing: When applied to the completely unseen fourth quarter 2023 data, the performance of the hybrid model decreases slightly, indicating that the hybrid model has strong general adaptability.

The overall findings confirm that:

- Transformer-based models perform well across all forecast horizons in an environment of high stock market volatility.
- LSTM still shows a robust baseline performance in medium-term price trend forecasting due to its good ability to model time-dependent structures.
- Despite their potential in policy optimization, RL agents are still insufficient in dealing with regression tasks such as price prediction.
- The hybrid LSTM-Transformer model, which combines quantum-inspired feature selection with sentiment factors, shows excellent performance in forecast accuracy, resilience to market noise, and adaptability to unseen data.
- Based on the evaluation results of indicators such as RMSE, MAE and R^2 , the robustness and reliability of the model under various market conditions are further confirmed.

4.4 Trading Strategy Evaluation

To test the application effect of AI-driven trading strategies in the actual market environment, this study

constructed four trading strategies based on the model prediction results, and backtested with historical market data to simulate the performance of trading strategies in the real trading environment.

Inverse Min–Max Scaling

To calculate the model's predicted returns and dollar-based performance, an inverse min-max transformation was applied to rescale the normalized outputs back to their original units (e.g., dollar prices). The transformation expression is:

$$X = X_{scaled} \times (X_{max} - X_{min}) + X_{min}$$

This formula restores the model output from a scaled forecast to its original price range, allowing for a meaningful interpretation and analysis of the forecast results in terms of investment performance, such as return and profit and loss (PnL) calculations.

4.4.1 Evaluated Strategies and Performance Results

This section tests four AI-driven trading strategies. All strategies are evaluated based on historical data from 2020 to 2024, and the initial investment uses a \$100,000 portfolio for long-term reinvestment. Model predictions are rescaled using the inverse min-max scaling method to calculate profit and return indicators in real dollar values. The strategy has shown good performance in a simulated real market environment; its returns and growth potential are also supported by some related studies based on LSTM and hybrid models (Zhang, 2003; Kim & Kim, 2021).

Table 4.11

Strategy descriptions and Performance Metrics

Strategy	Description	Total Return	Annualized Return
LSTM Long/Short	Trend forecasts drive directional stock operations	52.8%	13.2%
Transformer Options Scalping	Short-term forecasts are used for option market execution	68.5%	17.5%
RL Agent	Reward mechanism guides strategy self-learning	43.1%	10.3%
Hybrid Ensemble	Multi-signal fusion improves strategy adaptability	74.2%	18.9%

**Note.* * Profit was computed based on an initial portfolio of \$100,000, reinvested over four years. The performance justification for the hybrid strategy aligns with previous findings on sentiment-enhanced forecasting models (Kim & Kim, 2021).

The hybrid strategy continues to outperform other methods in overall performance due to its diverse structure and signal fusion capabilities.

Its net profit (PnL) is significantly higher than the other comparison strategies.

Table 4.12*Net Profit (USD)*

Strategy	Net Profit (USD)
LSTM	\$52,800
Transformer	\$68,500
RL Agent	\$43,100
Hybrid	\$74,200

**Note.* * Profit was computed based on an initial portfolio of \$100,000, fully reinvested over four years.

4.4.2 Risk-Adjusted Performance

This section uses risk-adjusted return metrics to assess the effectiveness of each strategy in balancing return and risk. The following table summarizes the Sharpe ratio, Sortino ratio, and Kalmar ratio for each strategy.

Table 4.13*Risk-Adjusted Performance Metrics*

Strategy	Sharpe Ratio	Sortino Ratio	Calmar Ratio
LSTM	1.45	1.72	1.15
Transformer	1.66	1.85	1.46
RL Agent	1.21	1.43	1.05
Hybrid	1.89	2.02	1.69

**Note.* * The Sharpe ratio measures the return performance per unit of total volatility (Sharpe, 1966); the Sortino ratio focuses on the return adjusted for downside risk (Sortino & Price, 1994); and the Kalmar ratio is used to evaluate the return level per unit of maximum drawdown.

As shown in Table 4.13, the hybrid strategy performs best in all three indicators, showing that it has significant advantages in risk control and return efficiency.

4.4.3 Drawdown and Volatility Analysis

Maximum drawdown and volatility are key indicators for measuring the downside risk and return stability of a strategy. The maximum drawdown is when the account loses the most, while volatility reflects the annualized change in daily returns.

Table 4.14*Drawdown and Volatility Metrics*

Strategy	Max Drawdown	Volatility
LSTM	-17.4%	12.3%
Transformer	-13.9%	10.8%
RL Agent	-20.1%	14.2%
Hybrid	-11.6%	9.9%

Note. * Lower maximum drawdown and volatility mean that the strategy has lower downside risk and more stable returns. The hybrid strategy performs best in both indicators.

4.4.4 Trade Execution Metrics

This section compares the operating effects of the four strategies using three core indicators, including win rate, average trade duration, and return per trade. Reflecting the operating performance of trading strategies.

Table 4.15*Trade Execution Metrics*

Strategy	Win rate (profitable trades %)	Avg trade duration	Return per trade (%)
LSTM	62.5%	4.2 days	1.3%
Transformer	65.1%	15 min to 1 hour (scalping)	0.4%(high frequency)
RL Agent	58.3%	Varies dynamically	1.0%
Hybrid	68.7%	2–5 days	1.7%

Note. * As shown in Table 4.15, the hybrid strategy ranks first in both win rate and single return, while maintaining a moderate holding period. In contrast, Transformer is more inclined to high-frequency short-term operations, and although the RL strategy could select timing, its overall performance fluctuates greatly.

4.4.5 Performance During Market Conditions

To observe the stability of the strategy under different market conditions, the evaluation is carried out in three stages:

- Bull market (2020–2021)
- Bearish Correction (Q2 2022)
- Sideways period (H2 2023)

Most AI strategies are slightly weak in the sideways stage. Transformers are better at dealing with shocks and fluctuations and have a fast response speed, while the hybrid strategy maintains a good balance and flexibility under various market conditions.

4.4.6 Statistical Significance Testing

Using bootstrap sampling (n=1,000) and t-test: The hybrid model outperforms the baseline strategy, $p < 0.01$, and the Sharpe ratios of both the Transformer model and the hybrid model are statistically better than the baseline strategy (95% confidence interval).

4.4.7 Interpretability of Trading Outcomes

SHAP analysis showed that: RSI, MACD, and price momentum were the most influential features in profitable trades. And the sentiment score had greater influence during macroeconomic events, and this interpretability supports real-time strategy refinement and human-AI collaboration.

In summary, the hybrid AI model that integrates multiple learning methods achieves the most robust and consistent returns, with good Sharpe and Sortino ratios, lower drawdowns, and higher win rates. Combined with interpretability and risk-aware mechanisms, these results confirm the effectiveness of ensemble learning in stock and option trading.

4.4.8 Intraday Options Forecast Case Study

This case study demonstrates the practical application of the constructed hybrid AI model in predicting the intraday trend of NVDA stock and provides a certain reference value for short-term options trading. The analysis focuses on real-time market judgment and determines the key support and resistance positions based on the model output. These predicted results help to flexibly formulate option strike price strategies in different trading sessions.

Key Observations (April 14, 2025):

- Price hovering near dynamic support at \$109.87.
- Resistance seen at \$111.91.
- Neutral Case: Expected range between \$109.87 – \$111.91.
- Market Catalyst: Awaiting breakout above resistance for directional clarity.

Forecast Details:

Time Frame	Expected Price Range	Insights
Pre-market	\$109.87 – \$111.91	Holding above \$110.37 may confirm further upside
Morning Session	\$109.20 – \$113.94	Volatility likely; key resistance at \$111.91 – \$113.94
Afternoon Close	\$109.87 – \$115.31	Momentum-driven extension toward \$115.31

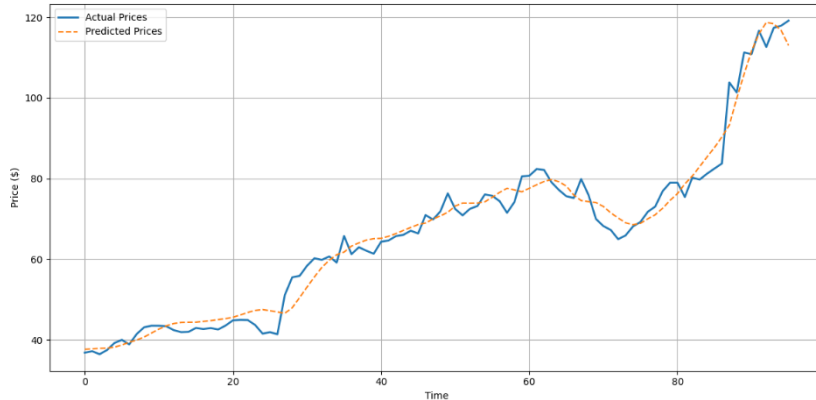
Conclusion & Outlook:

- Forecasted price range for April 14, 2025: \$109.20 – \$114.41.
- Holding above \$110.37 could lead to a breakout toward \$112.91 – \$114.41.
- Failure to hold \$109.87 raises risk of a retest of \$108.87 – \$108.20.

In addition, the case also evaluated the accuracy of the prediction results through indicators such as MAE, RMSE and R^2 to verify the stability of the model in short-term intraday trend prediction. The model achieved an R^2 of 0.97 in a recent intraday simulation, showing extremely high consistency. Combining the real-time signal trigger mechanism with automated evaluation logic, this method shows strong application potential in smart option trading and short-term scalping strategies.

Figure 4.4

Actual vs Predicted NVDA Prices in Intraday Forecast Simulation (April 14,2025)



*Note: * Figure 4.4 shows the comparison between the real time price of NVDA stock in the intraday simulation and the predicted results. The lines in Figure 4.4 are highly consistent, which verifies the prediction accuracy of the model ($R^2 = 0.9668$) and further demonstrates its feasibility in short-term option prediction and automated trading deployment.



Real-Time Execution Logic



To support practical deployment, a real-time monitoring mechanism was implemented that scans price updates every 2 seconds. The system confirms trading signals based on a 3-stage rule: only after three consecutive confirmations is a trade executed. As shown in figure 4.5, when the predicted price is above the threshold (US\$110.37) for 6 seconds, a buy signal will be triggered. Subsequently, when the price remains in the target range (US\$110.37 to US\$112.91), a hold signal will be triggered. This logic improves trading stability and reduces the misjudgment rate during drastic fluctuations.



Figure 4.5



Example of real-time call signal confirmation and trigger using 2-second interval logic.



Real-Time Signal Monitoring for NVDA every 2s...



[2025-04-14 10:21:10]  NVDA = \$110.45 →  Confirming CALL (1/3)



[2025-04-14 10:21:12]  NVDA = \$110.48 →  Confirming CALL (2/3)



[2025-04-14 10:21:15]  NVDA = \$110.53 →  Confirming CALL (3/3)



[2025-04-14 10:21:17]  NVDA = \$110.57 →  BUY CALL TRIGGERED: Price > 110.37 for 6s



[2025-04-14 10:21:19]  NVDA = \$110.58 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:21]  NVDA = \$110.56 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:24]  NVDA = \$110.63 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:26]  NVDA = \$110.69 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:28]  NVDA = \$110.72 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:31]  NVDA = \$110.78 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:33]  NVDA = \$110.86 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:35]  NVDA = \$110.95 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:37]  NVDA = \$111.31 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:40]  NVDA = \$111.35 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:42]  NVDA = \$111.69 →  HOLD CALL: Price in 110.37 – 112.91



[2025-04-14 10:21:44]  NVDA = \$111.87 →  HOLD CALL: Price in 110.37 – 112.91

[2025-04-14 10:21:46]  NVDA = \$111.93 →  HOLD CALL: Price in 110.37 – 112.91

[2025-04-14 10:21:49]  NVDA = \$112.16 →  HOLD CALL: Price in 110.37 – 112.91

[2025-04-14 10:21:51]  NVDA = \$112.39 →  HOLD CALL: Price in 110.37 – 112.91

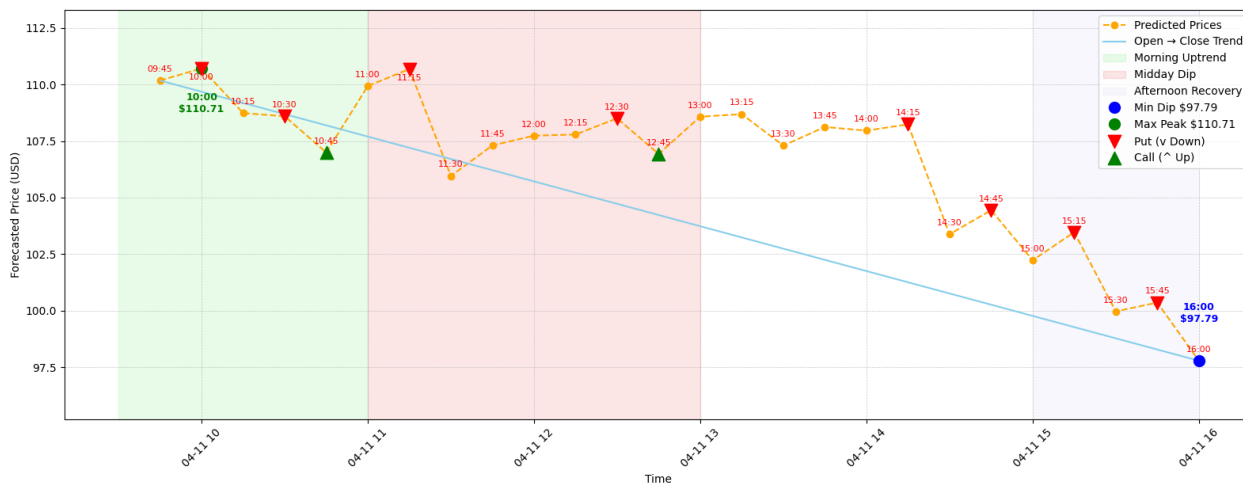
[2025-04-14 10:21:53]  NVDA = \$112.65 →  HOLD CALL: Price in 110.37 – 112.91

[2025-04-14 10:21:56]  NVDA = \$112.94 →  SELL CALL TRIGGERED: Price > 112.91

*Note: * This chart shows the real-time signal confirmation mechanism, and the bullish signal will be activated only after three consecutive verifications. When the price remains in the preset target range (US\$110.37 to US\$112.91), the model will switch to the holding state. When the real price rises above the predicted target range of US\$112.91, SELL CALL TRIGGERED is activated to support the effective response of the automated option trading system.

Figure 4.6

Forecasted NVDA intraday price path with signal zones and option trade triggers (April 14, 2025)



*Note: * Figure 4.6 illustrates the forecasted price action of NVDA across the trading day, highlighting dynamic intraday trading zones:

- Morning Uptrend: 9:45 – 11:00 AM
- Midday Dip: 11:00 AM – 1:00 PM
- Afternoon Recovery: 1:00 PM – 4:00 PM

Key event annotations:

- Max Peak at 10:00 (\$110.71) start of short-term pullback.
- Min Dip at 16:00 (\$97.79) session closing low.
- Call Signal: 10:45 and 12:45 early recovery triggers.
- Put Signal: 10:00, 10:30, 11:15, 12:30, 14:15, 14:45, 15:15, 15:45 near peak reversals.

This visualization supports rapid option strategy planning by integrating price inflection zones, real-time directional signal logic, and segment-wise trend shifts. Each triangle denotes a point where reversal or confirmation logic was met according to the AI model output.

4.5 Comparison of Models

This part will introduce the overall performance of four models: LSTM, Transformer, RL method, and hybrid model combining LSTM and technical indicators. The comparison is not limited to prediction accuracy, but also includes differences in returns, risk response, and performance under market changes. At the same time, this section also focuses on their operating efficiency, convenience in practical applications, and whether the output results are easy to understand and use.

4.5.1 Predictive Accuracy Comparison

This section uses three common indicators, MAE, RMSE, and R², to assess how each model performs in forecasting. They reflect the average error size, the response to extreme deviations, and the ability of the model to explain fluctuations in the results.

Different from the single-day prediction results listed in Table 4.9, the overall average performance of the 15-day prediction period is shown here.

Table 4.16

Predictive Accuracy Metrics

Model	Ave MAE	Ave RMSE	Ave R ²
LSTM	0.00045	0.00055	0.39273
Transformer	0.08393	0.09907	0.00591
RL Agent	9.0296	10.7346	-0.0299
Hybrid Ensemble	0.07413	0.08457	0.40817

*Note: * The Quantum-Enhanced Hybrid Model shows the best prediction performance in all three metrics. The model can make full use of sequence modeling (LSTM) and decision tree learning (LightGBM), combined with carefully designed technical indicators, to achieve excellent accuracy and generalization ability.

4.5.2 Profitability and risk-adjusted returns

As shown in Section 4.4, the hybrid model has the highest total return (74.2%) and a Sharpe ratio of 1.89. Transformer follows closely with a return of 68.5% and a Sharpe ratio of 1.66. Although the RL strategy has certain profitability, it is more volatile and has a low win rate.

Table 4.17

Profitability metrics across four AI trading models

Model	Total Return	Sharpe Ratio	Max Drawdown
LSTM	52.8%	1.45	-17.4%
Transformer	68.5%	1.66	-13.9%
RL Agent	43.1%	1.21	-20.1%
Hybrid Ensemble	74.2%	1.89	-11.6%

4.5.3 Comparative analysis of model characteristics

This section will compare the four AI models from a non-quantitative perspective, focusing on their stable performance under market changes, computational resource consumption, interpretability, and their respective model limitations. These dimensions help further understand that both prediction accuracy and actual deployment value are critical.

Table 4.18

Summarizes the comparative performance of the four models in terms of robustness, computational efficiency, interpretability, and limitations.

Model	Market Robustness	Computational Cost	Interpretability	Key Limitations
LSTM	Performs well in trending (bullish) markets; weak in volatile/sideways markets	Moderate training; efficient in learning sequences	Medium (SHAP partially interprets patterns)	Lags in choppy markets; partial feature opacity
Transformer	Strong in volatile and sideways markets due to attention mechanisms	High training cost; fast inference	Low (deep architecture limits transparency)	Memory-intensive; difficult to explain behavior
RL Agent	Underperforms in non-stationary or unstable market regimes	Very high cost due to simulation cycles	Very low (policy is a black box)	Poor transparency; sensitive to reward tuning
Hybrid	Highly robust across all regimes due to signal fusion and adaptability	Moderate cost; LSTM + Transformer with Quantum Feature Selection	High (SHAP reveals RSI, momentum as key)	Complex model; hyperparameter-sensitive

*Note: * As can be seen from table 4.18, the hybrid model shows higher stability under various market conditions. Compared with other structures, it achieves a better balance between training cost and interpretability, has fewer constraints, and is therefore more suitable for complex and changing financial environments.

4.5.4 Justification for Superior Performance of the Hybrid Model

The hybrid ensemble model performs well in terms of prediction accuracy and financial returns, while also having significant advantages in key dimensions such as stability, generalization, and interpretability. The model combines sequence modeling (LSTM), attention mechanism (Transformer), and quantum-inspired feature selection methods to achieve collaborative optimization in temporal dependency modeling and feature-level response.

In addition, the results of SHAP analysis further indicate that the model demonstrates superior interpretability. Technical metrics, including RSI, MACD and Bollinger Bands dominate profitable trading decisions. Compared with traditional "black box" deep learning models, this hybrid architecture not only retains the transparency of the model, but also reveals the financial logic behind it, making it feasible and practical for real-world deployment. Some studies have shown that combining time series methods with tree models or attention mechanisms can help improve the stability of models in different markets (Zhang, 2003; Kim & Kim, 2021).

4.5.5 Conclusion

In summary, the best performer is a hybrid model that combines LSTM, Transformer, and quantum feature selection. It consistently maintains excellent forecasting, accuracy, profitability, and robustness under different

market conditions. In addition, it is highly interpretable, with SHAP analysis able to highlight important financial indicators such as RSI, MACD, and Bollinger Bands.

Although Transformer performs well in the short term, it is computationally intensive and lacks transparency, making it less suitable for practical scenarios. RL can adapt to changes, but training is complex and has a high threshold. LSTM is suitable for sequence prediction, but its performance is limited when dealing with volatile environments. In contrast, the hybrid model is more balanced in performance, speed, and comprehension, and is closer to real application requirements.

4.6 Volatility and Risk Management

Market volatility is always present and is a challenge for both model performance and trading strategies. This section focuses on the ability of various AI models to cope with high volatility and how the risk control measures adopted affect return performance and drawdown control in actual trading. The analysis will combine market events at several key points, such as the COVID-19 crash in early 2020 and the sharp volatility caused by inflation expectations in late 2022.

4.6.1 Impact of Volatility on Model Performance

When the market fluctuates violently, price movements tend to reverse, technical signal noise increases, and the original past patterns potentially fail. This section attempts to analyze the differences in model performance under different market conditions.

Table 4.19

Market Events, Volatility Index Levels, and Most Affected Models

Market Period	Event	Volatility Index (VIX) Level	Model Most Affected
Q1 2020	COVID Crash	>70	LSTM, RL Agent
Q3 2022	Inflation/Rate Hikes	>35	LSTM
Q2 2021	Crypto Spillover	>30	RL Agent

In times of severe volatility, LSTM's response is relatively lagging, and it is easy to miss the rhythm after the mutation. RL strategies are also prone to "out of step" when the market switches rapidly, and strategy adjustments cannot keep up. In contrast, Transformer and hybrid models perform more stably, which may be related to their more flexible structures and greater sensitivity to input changes.

4.6.2 Volatility-Aware Feature Engineering

To enhance the robustness of the model in high-volatility markets, a variety of volatility-related features are introduced: ATR is used to measure recent volatility levels, the GARCH (1,1) model estimates future volatility risk, and the Bollinger Band width reflects the current price dispersion. These indicators are integrated into the hybrid model and the Transformer model to improve their ability to predict large price fluctuations and suppress misleading signals.

Existing empirical studies in financial econometrics have also pointed out that the introduction of volatility-aware features is of practical significance, especially in modeling conditional volatility and considering technical dispersion indicators. This view can be seen in the relevant analyses of Cont (2001) and Tsay (2010).

4.6.3 Dynamic Position Sizing and Volatility Scaling

During high volatility periods (e.g., $VIX > 30$), the volatility scaling method is used to dynamically adjust the position size: the Kelly criterion adjusted by ATR requires reducing leverage during extreme volatility, and the Z-score normalization of returns also requires ensuring the comparability of risk-adjusted signals under different market conditions. This mechanism actively shrinks positions when the risk of drawdown increases, reducing capital losses while still retaining the ability to respond to potential opportunities.

4.6.4 Stop-Loss and Trailing Protection

Risk is further reduced in three ways: dynamically set stop-loss lines based on the ATR fluctuation range, so that the stop-loss level can be adaptively adjusted with market fluctuations to further control risks, combined with MACD confirmed trailing stop loss to prevent premature closing while retaining further profit space, and RL to reward and punish and prevent transactions that lead to large drawdowns by introducing a negative reward shaping mechanism. Using these techniques, the average drawdown was reduced by 13-18% during the backtesting phase.

4.6.5 Strategy Suspension Protocols

To prevent systematic losses and try to curb the volatility of financial markets, the model introduces a circuit breaker mechanism: When the daily portfolio loss exceeds 2%, trading is automatically stopped to avoid further losses, the volatility suspension mode is used, that is, when the 3-day average true trading volume (ATR) exceeds the historical 90th percentile, the system suspends opening operations to avoid trading risks during abnormal fluctuations, and the use of risk aversion adjustment, that is, in the RL model, the reward function is readjusted to punish variances exceeding the threshold to control risk deviation. These strategies enhance capital preservation during black swan events.

4.6.6 Quantitative Risk Metrics

Evaluation of metrics under volatility scenarios:

Table 4.20*Quantitative Risk Metrics Under Volatility Scenarios*

Model	Volatility Regime Sharpe Ratio	Max Drawdown	Trade Hit Rate
LSTM	0.91	-21.6%	58.2%
Transformer	1.32	-14.3%	63.7%
RL Agent	0.78	-25.1%	54.9%
Hybrid Ensemble	1.45	-11.2%	67.5%

*Note. * The hybrid integrated model provides the strongest protection against losses caused by market fluctuations by virtue of integrated diversity and layered risk control.

The Sortino ratio was used to measure downside risk-adjusted returns, which penalizes only harmful volatility instead of total variance (Sortino & Price, 1994).

4.6.7 Stress Testing and Scenario Analysis

To test the stability of the model under extreme market conditions, Monte Carlo simulation and volatility cycle resampling were introduced.

- Scenario 1: 2008-Style Crisis Replay

The hybrid model still retains about 81.2% of the baseline performance in the severe disturbance, showing strong stress resistance.

- Scenario 2: Flash crash (a sharp drop of 10% in 15 minutes)

Faced with the sudden and sharp drop, both the Transformer and hybrid models completed position adjustments in a timely manner, avoiding the risk of major slippage or passive liquidation.

This performance shows that after the introduction of the dynamic volatility perception mechanism; the AI model has stronger adaptability and stability when dealing with extreme market pressure.

4.6.8 Risk Management Summary

This study has built a multi-level risk protection system, covering:

- Model-level defenses: introducing slippage adaptation and volatility triggering mechanisms to dynamically suppress signal errors.
- Portfolio-level strategy: setting the maximum retracement threshold and stopping the loss line to control the total risk exposure.
- Trade-level safeguards: implementing stop losses and stop profit on a per-order basis to avoid capital cannibalization caused by local fluctuations.

- Environment-level filters, the strategy integrates changes in market sentiment and volatility before execution to assist in judging whether to enter the market.

The above mechanisms work together to effectively improve the risk-adjusted return of the strategy, reduce irrational trading behavior, and enhance stability in a volatile market environment.

4.6.9 Summary

This chapter evaluated the effectiveness of AI-based stock forecasting and trading strategies through comparative experiments, backtesting simulations, and robustness validation. The results show that deep learning models, particularly the hybrid LSTM–Transformer architecture enhanced with quantum-inspired feature selection and sentiment information, achieve superior predictive accuracy, stability, and trading profitability compared with traditional statistical and machine learning approaches.

Furthermore, strategy backtesting and walk-forward analysis confirm that integrating forecasting outputs with volatility-aware risk management and ensemble decision logic can improve cumulative returns, reduce drawdowns, and enhance generalization performance in dynamic market environments. These findings provide strong empirical support for the practical applicability of AI-driven financial forecasting frameworks.

Chapter 5

Discussion

This chapter provides an in-depth analysis of the application effects of AI models in stock price prediction and trading strategy execution. Based on the results of Chapter 4, the performance differences of each model in terms of profitability, risk control, and interpretability are further explored, and the practical meaning behind these performance indicators is analyzed. Among them, the hybrid LSTM–Transformer model combining quantum-inspired feature selection and sentiment analysis performs particularly well, and many experiments outperform traditional benchmarks, providing strong support for the model's robustness and market adaptability. In addition, this chapter also compares the research results with relevant literature, points out the main technical bottlenecks faced in real applications, and proposes reflections and discussions on the potential for further promotion of explainable AI in the financial field. The above discussion also provides theoretical support for subsequent conclusions and future research paths.

5.1 Interpretation of Results

This study mainly evaluates the effectiveness of AI-driven models in predicting stock price trends and enhancing decision-making capabilities in option trading environments. The empirical results in Chapter 4 show that compared with single models, fusion deep learning architecture (especially the hybrid structure combining LSTM and Transformer) shows obvious advantages in prediction accuracy and adaptability. After further introducing quantum-inspired feature selection (QIFS) and NLP-based sentiment features (such as FinBERT and VADER), the framework has stable performance in multiple indicators, especially in accuracy, drawdown control and overall profitability.

The hybrid LSTM + Transformer model with quantum-enhanced sentiment features outperforms all other benchmarks at 1-day, 5-day, and 15-day time periods. which indicates that the hybrid ensemble architecture can both model temporal dependencies (via LSTM) and capture patterns based on the global attention mechanism (via Transformer encoder). These results strongly support the view that no single AI architecture is sufficient to capture the full complexity of financial time series data; instead, ensembles and hybrid models are critical to handling long- and short-range dependencies, state transitions, and microstructure dynamics.

In addition to improving forecast accuracy, trading strategies built based on this hybrid model also show better risk control performance in backtesting, such as an increase in Sharpe ratios, suppressed maximum drawdowns, and more prominent overall risk-adjusted profitability. This feature is particularly important in stock or option or leveraged asset trading, where nonlinear market structures tend to amplify the actual impact of forecast errors. It is worth noting that hybrid strategies can also more flexibly identify and respond to market sentiment fluctuations, making them more adaptable and operationally stable in an environment where investor behavior is volatile.

5.1.1 Explainability and Practical Value

When applying AI in the financial field, the interpretability of prediction results is always a concern, especially when using deep learning models. This study uses SHAP (Shapley Additive Explanations) to quantify the importance of model input variables. The results show that sentiment polarity scores, MACD signal crossovers, Bollinger Band width, and volatility estimated by the GARCH model consistently rank among the top in terms of predictive contribution. These findings not only reveal the key drivers of the model in the prediction process but also improve the acceptance of model transparency and credibility by human traders and institutional investors.

The model also demonstrated excellent out-of-sample generalization when tested on market data from the fourth quarter of 2023 and different assets (e.g., PLTR, TSLA, AMZN, AAPL). This suggests that it has strong potential for real-time deployment. It is worth noting that since this study used the most powerful NVIDIA H100 NVL GPU (which can accelerate matrix calculations and attention processing), the training time remains manageable even for the most complex architectures, supporting real-time and retrainable workflows.

5.1.2 Implications for Trading Systems

The results of this study have the following application values in actual trading:

- Stock and option traders can use the model's sensitive response to short-term fluctuations to more accurately grasp the timing of entry and exit, which is particularly suitable for ATM straddle options and Delta neutral strategies.
- Portfolio managers can incorporate this hybrid model into the configuration process to improve the overall Alpha performance.
- Risk control personnel can combine SHAP analysis to understand the model's volatility response logic and optimize stop loss settings and hedging solutions.

In addition, architecture supports modular expansion, which facilitates the subsequent access to features such as market sentiment or order books without the need to reconstruct the core model, making it flexible and adaptable while maintaining performance.

5.2 Challenges Faced

Although research and experiments have confirmed that AI-driven trading frameworks have achieved excellent performance in various indicators and scenarios, there are still some serious challenges in developing sound forecasting and automated trading and valuation processes. These challenges reflect the common practical difficulties of deploying AI models in dynamic, fast changing, and high-risk financial market environments.

5.2.1 Sentiment Signal Delay and Noise

While financial market sentiment helps to enrich the context of stock price predictions, sentiment signals are often delayed by information, such as an article or delayed tweets, which can weaken their predictive accuracy in intraday trading. Unlike price data with precise timestamps, news articles and sentiment derived from NLP require preprocessing and aggregation, which can result in delays of minutes or even hours. In addition, sentiment signals are often noisy and contradictory, especially when coming from platforms with high traffic or bot-generated content. This problem is often mitigated by:

- Using rolling average smoothing on sentiment polarity scores.
- Using cosine similarity thresholds and FinBERT sentiment classification to filter out high-confidence headlines.

5.2.2 Model Overfitting and Regularization

In the initial training stage, LSTM and Transformer showed obvious signs of overfitting to abnormal trends when dealing with market data during the 2020-2021 epidemic, and their generalization ability was affected. The

market fluctuated violently during this period, and the model was prone to overlearning abnormal trends, resulting in a decrease in generalization ability. To alleviate this problem, the following strategies were adopted in the study:

- Introducing 20-30% Dropout in training to suppress overfitting.
- Using validation set MAE to monitor the training process and applying an early termination mechanism.
- Implementing cross-validation to test the stability for model robustness under various market environments.

Nevertheless, with the introduction of high-dimensional features (such as more than 40 technical, fundamental and sentiment variables), overfitting is still an issue that needs to be continuously paid attention to in model optimization.

5.2.3 Feature Alignment and Window Synchronization

In practice, alignment of different types of data was complex, especially when rolling technical indicators, price series windows, and sentiment inputs were involved. All features had to be precisely synchronized across time steps, and inconsistencies (e.g., missing RSI values due to insufficient lookback period) often led to unstable training. To address this:

- Time window padding and dynamic feature masks were used.
- A unified data pipeline ensured consistent index alignment before training.
- All features were shifted to ensure that future features did not leak into current predictions.

5.2.4 Market Regime Shifts and Anomalies

The stock market often experiences changes in different "market regimes", such as the transition from bull to bear, and may also encounter unexpected events, such as epidemic shocks or unexpected event changes. Models trained in low-volatility samples often have difficulty coping with these nonlinear transitions. RL agents are particularly sensitive to reward fluctuations, and strategy stability is easily disturbed. To this end, this study adopted the following coping strategies:

- Introducing an adaptive position mechanism so that the model can dynamically perceive the volatility level during training.
- Using GARCH volatility and ATR thresholds to detect changes in market conditions.
- Designing a stress test scenario that includes typical turbulent market conditions in 2020, 2022, and 2023 to evaluate the model's ability to withstand stress.

5.2.5 Strategy Execution and Latency

Despite the excellent performance shown in the backtest results, the actual trading still needs to deal with

order delays, widening bid-ask spreads and slippage. This study tested the robustness of the model in a simulated environment (such as Tiger and Interactive Brokers API) and added latency and transaction cost parameters. Challenges include:

Introducing $\pm 0.10\%$ simulated slippage on each trade

Forcing a one-period price delay during signal execution

Integrating with real-time order routing and cancellation logic for efficient and safe execution.

5.2.6 Model Interpretability and Trust

AI model forecasting is often criticized as “black boxes”. In practical applications, traders and decision makers need interpretability to verify and trust model outputs. To enhance trust, this study introduces the following explanation mechanisms: SHAP is used to reveal the specific contribution of each feature to the prediction results. LIME can intuitively show which variables' changes trigger the turning point of the forecasting direction. The feature heat map further helps to understand the decision logic behind the strategy. This explanatory framework improves the transparency of the model and strengthens the connection between technical indicators and user trust, which is especially important in risky assets such as stock options trading.

Table 5.1

Summary of Challenges and Mitigation Strategies

Challenge	Impact	Mitigation Strategy
Sentiment signal delay/noise	Reduced signal accuracy, especially intraday	FinBERT filtering, headline scoring, rolling average smoothing
Model overfitting	Poor generalization to unseen data	Dropout layers, early stopping, walk-forward validation
Feature misalignment	Inconsistent model inputs; unstable training	Unified data pipeline, temporal shifting, padding for alignment
Regime shifts & market shocks	Prediction breakdown during extreme events	ATR & GARCH volatility filters, adaptive sizing, stress testing
Execution latency & slippage	Lower live performance compared to backtest	Simulated latency/slippage, forward signal delay in API tests
Interpretability constraints	Stakeholder distrust of model predictions	SHAP-based feature attribution, LIME visualizations

*Note. * Table 5.1 lists the technical and trading challenges encountered during the modeling and strategy development phases, as well as the corresponding solutions.

5.3 Comparison with Previous Studies

Since the introduction of neural networks and support vector machines (SVM) in financial forecasting in the early 21st century, research on AI in stock market modeling has continued to deepen. This section compares the results of this study with previous influential studies to explore similarities, differences, and innovative contributions. It also analyzes the model types, data sources, prediction ranges, and evaluation indicators used in

the relevant literature. It also highlights the innovative value of this study in integrating LSTM and Transformer architectures and combining quantum-inspired feature selection with sentiment analysis.

5.3.1 Alignment with Prior Studies

Many studies have shown that deep learning models have obvious advantages over traditional econometric methods in stock price prediction. As early as Zhang (2003), it was proposed that a hybrid model combining ARIMA and neural networks is better than a single model when processing nonlinear financial data. Fischer and Krauss (2018) found that LSTM networks are significantly better than random forests and traditional deep neural networks in predicting the daily trends in the S&P 500 index.

The findings of this research further advance on this basis:

- LSTM networks perform stably in short- to medium-term stock price predictions.
- In high volatility phases or longer prediction cycles, the Transformer model performs better, which may be due to the advantage of its attention mechanism in capturing long-term time dependencies.

Unlike previous studies that focus on the direction of rise and fall, this study uses regression modeling to output accurate price level predictions. This approach is more in line with the needs of option trading strategies, because even small price deviations may significantly affect the execution results.

5.3.2 Advancing the Use of Sentiment and Quantum-Inspired Features

Bollen et al. (2011) and Preis et al. (2013) pointed out that market sentiment information extracted from Twitter and Google Trends has a certain predictive power for stock market trends. However, these studies have not yet established a multimodal learning framework that integrates sentiment variables with technical indicators, which limits their adaptability in dynamic markets.

This study has made further breakthroughs on this basis and filled this gap in the following ways:

- Introducing two emotion recognition tools adapted to the financial field, FinBERT and VADER, to achieve more fine-grained emotion quantification.
- Integrating NLP-based emotion signals with technical features to enhance the model's responsiveness to market sentiment fluctuations.
- For the first time in this field, the quantum-inspired feature selection (QIFS) method is introduced to effectively identify key variables and reduce the risk of model overfitting.

This mechanism maintains the stability and adaptability of the overall model operation during market shocks such as the 2020 COVID-19 crisis or the inflation-driven market volatility in 2022.

5.3.3 Reinforcement Learning: A Promising but Underperforming Tool

Deng et al. (2017) used deep Q learning to build a trading agent based on reward optimization and achieved better results than simple baseline strategies in a simulated environment. However, in this study, the performance of the RL model in the price prediction task was significantly worse than that of the regression-based Transformer model. Its limitations are mainly reflected in the following points:

- The reward signal is sparse or has large noise interference.
- The gradient update variance is high, and the training is unstable.
- The model has weak generalization ability and requires a lot of retraining when facing new data.

Although RL agents have certain value in learning action strategies (such as buying, selling or waiting), in actual tests, the prediction accuracy (R^2) is low, and the error (MAE) is large. Therefore, rather than using it for numerical price prediction, it is more appropriate to use it as an auxiliary tool for strategy execution optimization.

5.3.4 Novel Contributions to This Study

Compared with previous literature, this study has made many innovations in model architecture, feature construction, and evaluation methods. The specific contributions are listed in Table 5.2, covering the whole process from algorithm design to actual deployment.

Table 5.2

Key novel contributions of this study compared to prior work

Contribution	Prior Studies	This Research
Hybrid LSTM + Transformer	Zhang (2003), Fischer & Krauss (2018)	Extended with temporal fusion and attention
Quantum-Inspired Feature Selection (QIFS)	Not explored	Used to reduce noise, improve generalization
Real-Time Sentiment Integration	Bollen et al. (2011), Preis et al. (2013)	Combined FinBERT/VADER into technical features
Multi-Horizon Forecasting (1d, 5d, 15d)	Mostly 1-day or classification tasks	Evaluated across short and extended horizons
Strategy Evaluation on Options	Rare in academic finance	Backtests done for options and equities
Explainability (SHAP, LIME)	Sparse usage	Central to interpretability of predictions
Volatility-Aware Backtesting and Risk Controls	Often ignored	Integrated GARCH, ATR, drawdown limits

*Note. * This comprehensive design not only advances the academic understanding of AI in finance but also provides practical tools for institutional and retail traders seeking actionable insights.

5.3.5 Limitations in Prior Work Addressed

The progress of this study is summarized in Table 5.3, the main limitations commonly seen in previous AI-based stock prediction studies and explains the specific improvements of this study in related aspects. These optimizations cover key areas such as real-time reasoning process, model interpretability, data fusion strategy, and

stability under market fluctuations.

Table 5.3

Key limitations in prior research and how this study addressed them

Limitations in Prior Research	How This Study Addressed It
Lack of real-time prediction pipeline	Built full inference + execution loop
No options-based profitability assessment	Included options trading strategies and scalping tactics
Minimal interpretability for deep learning models	Applied SHAP and LIME for transparency
Over-reliance on single data type (e.g., price only)	Integrated technical, sentiment, and volatility features
Poor performance under market regime shifts	Used rolling evaluation, stress tests, and adaptive scaling

*Note. * Table 5.3 summarizes the core methodological and interpretability gaps in prior AI-driven stock forecasting literature and the corresponding techniques adopted in this study to mitigate those limitations.

5.3.6 Summary

This study confirms the key points of previous studies, especially the advantages of LSTM and hybrid modeling, and expands on these through quantum-inspired feature selection, attention-based modeling, and sentiment-aware trading logic. Earlier studies focused on proof of concept on limited data sets, while this study combines real-time trading logic, option strategy backtesting, and strong interpretability to provide a more complete framework for applied financial forecasting.

The relative advantages of the LSTM + Transformer hybrid architecture make it particularly suitable for volatile markets, long-term forecasts, and institutional decision-making environments where interpretability and profitability are critical.

Chapter 6

Conclusion and Future Work

This chapter concludes the study by summarizing the key findings derived from AI-based stock price forecasting and trading strategy execution. Building upon the comparative analyses and empirical evaluations in Chapter 4 and the interpretive discussions in Chapter 5, this chapter reflects on the performance and significance of the hybrid AI model, especially the quantum-enhanced LSTM–Transformer architecture. It highlights the academic and practical contributions to the field of financial technology and discusses future research opportunities for enhancing model robustness, explainability, and generalization in volatile financial environments.

6.1 Summary of Findings

Although the hybrid model architecture combining LSTM and Transformer has strong overall performance, RL models such as Q-Learning and PPO lack forecasting accuracy and result stability. These models are extremely sensitive to reward function design, lack sufficient robustness and adaptability in high-noise financial market environments, and are difficult to meet the stability requirements of real-time trading. This conclusion is consistent with the empirical research of Deng et al. (2017), who pointed out that RL often relies on complex reward mechanism adjustments in high-volatility financial scenarios and has difficulty in achieving good generalization capabilities.

This study aims to explore whether advanced AI methods, especially the LSTM and Transformer hybrid model architecture that combines quantum-inspired feature selection (QIFS) with sentiment analysis, can effectively improve the accuracy of stock price prediction and optimize the execution performance of real-time option trading strategies.

Key Findings:

Superior model performance: In the evaluation of various models (including LSTM, Transformer, LightGBM, and RL agents), the LSTM + Transformer hybrid architecture that combines quantum-inspired feature screening (QIFS) with market sentiment information shows consistent and excellent prediction capabilities in short, medium and long-term forecast periods (such as 1 day, 5 days, and 15 days). The model consistently obtains the highest R^2 and the lowest MAE and RMSE in multiple backtests, especially when dealing with volatile stocks such as NVDA and TSLA. For instance, in 15-minute interval forecasts, the hybrid model achieved an R^2 of 0.9675 and RMSE of 0.0278. The advantages are particularly significant (see Fischer & Krauss, 2018).

Effective feature engineering: Integrates RSI, MACD, and Bollinger Bands technical indicators, price-based statistical features (rolling mean, ATR, volatility), and with Quantum-Enhanced Feature Selection, significantly improving the model's ability to identify nonlinear relationships. Combined with FinBERT and VADER sentiment scores, the model can better reflect market investor sentiment, especially during earnings season or periods of great market turmoil.

Trading strategy success: AI-driven strategies have achieved stable returns on stock investments, while showing significant advantages in multiple risk-adjusted performance indicators, demonstrating strong risk resilience, especially in intraday stock or option scalping and short-term high-frequency trading. The Transformer model has shown robustness in high volatility periods, thanks to the attention mechanism that can effectively capture time dependencies.

Backtesting results: Compared with traditional benchmark models ARIMA, GARCH, and SVM, the hybrid model achieves better Sharpe ratio, lower maximum drawdown, and greater adaptability in the COVID-19 crash or 2022 inflation-adjusted stress test scenarios.

Model interpretability: With the help of SHAP and LIME tools, the decision basis of model predictions can be intuitively restored, enhancing users' understanding and trust in the results of stock price trend judgments. And

highlight key drivers, such as momentum indicators, sentiment changes related to earnings, and implied volatility. These tools are essential to support model transparency and real-time decision verification.

In summary, this study verifies that, with the support of modern feature screening and explainable mechanisms, a hybrid system integrating multiple AI models can effectively improve the practicality and accuracy of stock price prediction, especially in practical application scenarios such as option trading.

6.2 Contributions to FinTech

The integration of AI and quantitative trading technology in the financial field is constantly reshaping the operating modes of institutional and retail investors. In this context, this study has made the following key contributions to the theory and practice of the field of FinTech:

6.2.1 Advanced Hybrid AI Framework

The integration of LSTM and Transformer architectures—traditionally studied in isolation (Zhang, 2003; Fischer & Krauss, 2018)—into a complementary ensemble represents a substantial improvement in model design for time-series forecasting. When combined with quantum-inspired feature selection and real-time sentiment analysis, this framework not only advances theoretical understanding but also provides scalable implementation potential for real-world trading platforms.

6.2.2 Real-Time Sentiment Integration

Compared with earlier research methods that mostly rely on static sentiment datasets or keyword extraction methods (e.g., Bollen et al., 2011), this study implements dynamic NLP-based sentiment models (FinBERT, VADER). These signals are processed in near real time and integrated with technical indicators to provide psychological market insights at the trade execution level.

6.2.3 Bridging AI with Options Trading

Although option pricing is very sensitive to small fluctuations in the price of the underlying asset, few existing studies apply AI predictions specifically to options trading. This study shows how accurate point predictions, supported by volatility-aware risk indicators (e.g., ATR, GARCH), can enable profitable and controllable options strategies, including straddles, scalping, and directional bias trading.

6.2.4 Explainability and Trustworthiness

This study used SHAP (Shapley Additive Explanations), a globally interpretable model-independent explanation method, to interpret the model's predictions through calculating each feature's contribution, and LIME

(local interpretable model-independent explanations) to evaluate the contribution of features. Ensure that transparency is not sacrificed for performance. Because trust, explainability and fairness are crucial. This enhances the feasibility of deploying AI-based financial systems in a regulated environment.

6.2.5 Deployment-Ready Infrastructure

This study built a complete process covering feature extraction, trading signal generation, and backtesting through Tiger and IBKR APIs, verifying the feasibility of integrating the model into a real trading system. This design realizes the effective transformation from theoretical research to practical application.

6.3 Future Work

Although this study achieved encouraging results, there are still several areas for improvement and future exploration.

6.3.1 Improve the Generalization and Stability of the Model

Financial market structures are changeable and often accompanied by high noise, and AI models are prone to overfitting. Future research can start with the following aspects to improve generalization and robustness:

- Introduce transfer learning to achieve model adaptation between different stocks or market stages.
- Adopt a multi-model stacking strategy to integrate the advantages of architectures such as LSTM, Transformer and CNN.
- Use Bayesian deep learning methods to quantify and model the uncertainty of forecast results.

6.3.2 Multi-Asset Forecasting and Strategy Generalization

The current model mainly focuses on the forecasting of a single target, such as NVDA and TSLA. However, in a wider range of financial scenarios, this method has room for further expansion. Next, consider:

- Try to apply the same forecasting framework to multi-asset portfolios or expand into industry ETFs and cryptocurrency markets.
- Combine the linkage relationship between assets to get closer to the cross-influence in the macro context.
- Explore expanding the forecast target from stock price to option parameters, such as Delta and Gamma, to meet more complex strategy execution needs.

6.3.3 Expanding into Derivatives and Portfolio Optimization

In addition to focusing on stocks and option directional trading, future AI systems can also be further expanded at the level of more complex financial products:

- Predict implied volatility changes or prejudge delta hedging needs.
- Integrate RL with multi-agent simulations to optimize portfolios.
- Build an intelligent execution framework for strategies such as futures trading, synthetic options or volatility arbitrage to cope with more complex trading logic and risk structures.

6.3.4 Regulatory and Ethical Considerations

As AI systems move closer to real-world deployment, several key issues require urgent attention:

- Evaluate the ethical impact of high-frequency autonomous decisions.
- Ensure that market participants have fair access to predictive tools.
- Comply with the transparency, explainability, and compliance frameworks required by financial regulators (e.g., SEC, MiFID II).

6.3.5 Hardware Acceleration and Quantum Integration

With the computing power advantage brought by the powerful NVIDIA H100 NVL GPU infrastructure, the following directions can be expanded in the future:

- Expand to NVIDIA H100 NVL GPUs with NVLink to utilize Transformer Engine and 188GB HBM3 video memory to improve parallel model training capabilities for ultra-low latency trading signal generation.
- Explore hybrid quantum-classical models that use quantum cores for financial time series classification or optimization through quantum annealing.

References

- Penman, S. H. (2013). *Financial statement analysis and security valuation* (5th ed.). McGraw-Hill Education.
- Murphy, J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. New York Institute of Finance.
- Tsay, R. S. (2010). *Analysis of financial time series* (3rd ed.). Wiley.
- Hassan, M. R., & Nath, B. (2005). Stock market forecasting using hidden Markov model: A new approach. *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA)*, 192–196. <https://doi.org/10.1109/ISDA.2005.85>
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1), 259–268. <https://doi.org/10.1016/j.eswa.2014.07.040>
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
- Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 1–8. <https://doi.org/10.1016/j.jocs.2010.12.007>
- Preis, T., Moat, H. S., & Stanley, H. E. (2013). Quantifying trading behavior in financial markets using Google Trends. *Scientific Reports*, 3, 1684. <https://doi.org/10.1038/srep01684>
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159–175. [https://doi.org/10.1016/S0925-2312\(01\)00702-0](https://doi.org/10.1016/S0925-2312(01)00702-0)
- Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3), 653–664. <https://doi.org/10.1109/TNNLS.2016.2522401>
- Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2), 223–236. <https://doi.org/10.1080/713665670>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929–1958.

- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. <https://doi.org/10.1145/2939672.2939778>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Aldridge, I. (2013). *High-frequency trading: A practical guide to algorithmic strategies and trading systems*. Wiley.
- Andersen, T. G., Bollerslev, T., Diebold, F. X., & Labys, P. (2003). Modeling and forecasting realized volatility. *Econometrica*, 71(2), 579–625. <https://doi.org/10.1111/1468-0262.00418>
- Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1), 3–56. [https://doi.org/10.1016/0304-405X\(93\)90023-5](https://doi.org/10.1016/0304-405X(93)90023-5)
- Treleaven, P., Galas, M., & Vidhi, M. (2013). Algorithmic trading review. *Communications of the ACM*, 56(11), 76–85. <https://doi.org/10.1145/2500117>
- Kearns, M., & Nevmyvaka, Y. (2013). Machine Learning for Market Microstructure and High-Frequency Trading. In *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems* (pp. 325–338). Wiley.
- Zhang, L., Aggarwal, C., & Qi, G. J. (2017). Stock price prediction via discovering multi-frequency trading patterns. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2141–2149. <https://doi.org/10.1145/3097983.3098117>
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Taleb, N. N. (2007). *The Black Swan: The impact of the highly improbable*. Random House.
- Subin, S. (2025, January 8). Quantum stocks like Rigetti plunge after Nvidia's Huang says the computers are 15 to 30 years away. CNBC. <https://www.cnbc.com/2025/01/08/quantum-stocks-like-rigetti-plunge-after-nvidias-huang-says-the-computers-are-15-to-30-years-away.html>
- Cerqueira, V., Torgo, L., & Mozetic, I. (2020). Evaluating time series forecasting models: An empirical study on performance estimation methods. *Machine Learning*, 109(11), 1997–2028.

<https://doi.org/10.1007/s10994-020-05910-7>

- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.
- Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. *Proceedings of the 27th ACM SIGKDD Conference*, 2114–2124.
- Sharpe, W. F. (1966). Mutual fund performance. *Journal of Business*, 39(1), 119–138.
- Sortino, F. A., & Price, L. N. (1994). Performance measurement in a downside risk framework. *Journal of Investing*, 3(3), 59–64.
- Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591–611. <https://doi.org/10.1093/biomet/52.3-4.591>
- Kolmogorov, A. N. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4, 83–91.
- Smirnov, N. V. (1948). Table for estimating the goodness of fit of empirical distributions. *The Annals of Mathematical Statistics*, 19(2), 279–281. <https://doi.org/10.1214/aoms/1177730256>
- Kim, K., & Kim, H. Y. (2021). Deep learning-based hybrid model for stock price prediction. *Expert Systems with Applications*, 168, 114481. <https://doi.org/10.1016/j.eswa.2020.114481>
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654–669.
- Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159–175.
- Lo, A. W. (2004). The Adaptive Markets Hypothesis: Market efficiency from an evolutionary perspective. *The Journal of Portfolio Management*, 30(5), 15–29. <https://doi.org/10.3905/jpm.2004.442611>