DEEP LEARNING METHODS FOR STOCK PREDICTION

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisor

Prof. Jian Yu

Associate Prof. Sira Yongchareon

2025

By

Lei Zhou

School of Engineering, Computer and Mathematical Science

Abstract

Stock market prediction is a challenging task due to the inherent volatility, non-stationarity, and complex interdependencies within financial markets. Although deep learning models have demonstrated strong capabilities in modeling time-series and textual data, their practical accuracy and ability to generalize across varying market conditions remain limited. A key limitation is the difficulty of effectively capturing the dynamic and evolving relationships among stocks, particularly those driven by both market behavior and external textual information. Moreover, the lack of a unified framework capable of integrating heterogeneous modalities such as numerical indicators, candlestick chart images, and semantic insights from financial news significantly impairs the robustness of models when faced with diverse market conditions.

The reasons for these challenges include: First, traditional models often fail to capture long-range dependencies and complex topological structures among stocks. Second, there is limited integration of multimodal data sources including technical indicators, price patterns, and financial news. Third, existing approaches lack effective mechanisms to uncover latent inter-stock relationships using textual signals from sources such as news articles, social media, or analyst reports such relationships could be more effectively modeled through motif-based graph structures. Fourth, the interpretability of current deep learning models remains limited, impeding their adoption in high-stakes financial decision-making scenarios. To address these challenges, this thesis proposes three distinct yet complementary modeling frameworks, each designed to target specific

aspects of the stock prediction problem through multimodal data fusion and dynamic relational modeling.

This thesis aims to advance the field of stock market prediction by proposing a suite of models that integrate temporal, textual, image-based, and graph structural information. In particular, this research addresses the aforementioned challenges through the combined application of deep learning and graph-based modeling. The proposed approaches introduces novel frameworks that embed semantic, structural, and temporal features into unified architectures, enhancing predictive robustness and interpretability. Given the persistent challenges in financial forecasting, there is a clear need for a comprehensive framework capable of leveraging multimodal information, encoding meaningful inter-stock and stock-news relationships, and improving the interpretability and accuracy of predictions in a scalable manner.The thesis follows three key research questions.

The first part of this thesis addresses the challenge of capturing hidden associations between stocks and financial news, which are often overlooked by traditional models. These implicit relationships carry valuable signals that can influence stock price movements. To resolve this, we propose a Motif-based Graph Convolutional Network (MGCN) that constructs motif graphs by linking stocks and news entities based on semantic patterns extracted from financial texts. A Transformer encoder is further employed to refine both price and text features, enabling the model to capture long-range dependencies.. Experimental on the S&P 500 dataset show that this model effectively integrates textual and temporal signals, achieving improved prediction accuracy over standard baselines.

The second part of this thesis tackles the difficulty of integrating diverse financial signals such as temporal patterns, visual structures, and contextual sentiment within a unified predictive framework. Most existing models rely on a single modality, limiting their ability to generalize across dynamic market conditions. To address this, we propose a hybrid deep learning framework that fuses three complementary modalities. A Linear Transformer processes historical stock prices and technical indicators to extract temporal dependencies. Candlestick chart images are encoded via a CNN to capture spatial features. Concurrently, a Large Language Model (LLM) generates daily textual analyses, which are embedded using FinBERT to represent semantic sentiment. These multimodal features are combined through a feedforward network to produce final predictions. Empirical results demonstrate that this integrated approach significantly improves forecasting accuracy compared to unimodal baselines.

The final part of this thesis addresses the challenge of capturing dynamic inter-stock relationships that are influenced by textual semantics but are often overlooked by conventional models. Most existing graph-based approaches rely on static price correlations, ignoring latent sentiment-driven connections revealed in financial discourse.To overcome this, we introduce an LLM-Augmented Enhanced Graph Transformer framework. In this approach, a large language model (LLM) generates concise daily analyses from technical indicators, which are then embedded via FinBERT to reflect semantic context. These embeddings are used to construct a graph where edges represent semantic similarity between stocks, enabling a Graph Transformer to model nuanced relational dependencies. Experimental results on the S&P 500 dataset show that this framework significantly outperforms time-series baselines (e.g., LSTM, Transformer, Informer) and prior graph models (e.g., GCN, GAT), delivering superior accuracy and interpretability in stock movement prediction.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Publications

The following publications have resulted from the research presented in this dissertation:

1. Zhou, L., Zhang, Y., Yu, J., Yongchareon, S., Madanian, S., & Wang, G. (2024). *A Short Survey on Graph Neural Networks Based Stock Market Prediction Models*. In *Proceedings of the IARIA Conference*, Venice, Italy, April 14–18, 2024, pp. 10–16. ISBN: 978-1-68558-147-3, ISSN: 2308-3972. © IARIA 2024.

2. Zhou, L., Yu, J., Zhang, Y., Wang, N., Yu, J., Wang, G., & Zheng, X. (2024). *A Motif-based Graph Convolution Network for Stock Trend Prediction*. In *Proceedings of the 31st International Conference on Neural Information Processing (ICONIP 2024)*, pages 1–15.

3. Zhou, L., Zhang, Y., Yu, J., Wang, G., Liu, Z., Yongchareon, S., & Wang, N. (2025). *LLM-Augmented Linear Transformer–CNN for Enhanced Stock Price Prediction*. Mathematics, 13(3), 487.

4. Zhou, L., Chen, Y., Yu, J., Yongchareon, S., Ruan, J., & Zhang, Y. (2025). *LLM-Augmented Enhanced Graph Transformer for Stock Movement Prediction*. In *Proceedings of the IEEE Region 10 Symposium (TENSYMP 2025)*, University of Canterbury, Christchurch, New Zealand, July 7–9, 2025. (Accepted).

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Jian Yu, for your invaluable guidance, encouragement, and unwavering support throughout my academic journey. Your mentorship has not only shaped my approach to research but also inspired my pursuit of excellence. Your belief in my potential was the driving force behind my progress, and this thesis would not have been possible without your insight and dedication.

I am also sincerely thankful to Yuqi Zhang and Nancy Wang for their kind assistance with numerous tasks, as well as for their constructive feedback and suggestions, which greatly improved the quality of my work. I also would like to thank Yilong Chen for his support during this work.

Finally, I wish to express my heartfelt appreciation to my family. To my wife, thank you for your unwavering emotional and material support, and for always standing by my side through every challenge. To my parents, thank you for your unconditional love, patience, and for always believing in me. Your faith and encouragement have been the foundation of all my achievements. I love you all, forever.

Chapter 1

Introduction

This chapter introduces the motivation, background, and core challenges of stock market prediction in the era of deep learning. It reviews traditional and modern methodologies including statistical models, machine learning, and graph-based approaches and defines the three research questions that guide this dissertation.

1.1 Background

The fluctuations in stock market prices reflect fundamental industry and economic conditions, influencing investor decisions and consequently shaping broader economic trends (Shiller, 2009). Accurate prediction of future stock price trends can significantly assist investors in making more informed decisions, minimizing potential losses, and maximizing potential gains (W. Huang, Nakamori & Wang, 2005).

With the rapid advancement of AI technology, stock prediction has entered a new era where traditional statistical models are increasingly being complemented or replaced by intelligent learning systems. Deep learning methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated significant success in capturing complex temporal patterns in stock price data (Fischer & Krauss, 2018). More recently, the integration of graph-based learning techniques has attracted considerable attention, enabling models to exploit the intrinsic relationships between stocks through graph structures (F. Feng et al., 2019).

In the field of stock market analysis, representing inter-stock relationships as a graph structure has become a powerful modeling approach. In such a graph, each node denotes an individual stock, while edges capture meaningful connections such as industry categorization, supply chain dependencies, or historical price correlations (Saha, Gao & Gerlach, 2022).

Graph Neural Networks (GNNs), along with variants such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs), have been widely adopted to model the co-movements among stocks and dependencies across sectors. These models effectively capture the complex interactions and relational patterns that exist among financial entities, leading to improved prediction accuracy (M. Patel, Jariwala & Chattopadhyay, 2024).

Moreover, recent advancements introduce the concept of graph motifs frequently occurring and semantically meaningful subgraph patterns to further enhance the expressiveness of graph-based models. The detection and utilization of motifs have shown to reveal micro-structural patterns within the financial market, providing deeper insights into temporal and relational dynamics (Ma, Li, Feng, Fang & Zhang, 2023).

The integration of these advanced graph-based techniques enables a more comprehensive understanding of stock behaviors through relational learning, supporting more informed forecasting and investment strategies.

At the forefront of modern AI research, Large Language Models (LLMs), which are fundamentally built upon Transformer architectures, have emerged as state-of-the-art tools for a wide range of tasks, including stock prediction. LLMs such as ChatGPT and DeepSeek exhibit remarkable capabilities in understanding context, generating coherent textual content, and reasoning over complex inputs. Their ability to capture long-range dependencies and to integrate multi-modal information including financial news, social sentiment, and structured time-series data makes them particularly well-suited for financial forecasting (Q. Zhang et al., 2022; N. Wang, Yang & Wang, 2023). The integration of LLMs into stock prediction frameworks represents a significant step forward, offering enhanced interpretability, contextual awareness, and predictive accuracy.

1.2 Challenges in Stock Prediction

Stock market prediction remains one of the most intricate and consequential tasks in quantitative finance. Despite the development of increasingly sophisticated modeling techniques, accurately forecasting price movements particularly in diversified indices such as the S&P 500 continues to pose substantial challenges due to the stochastic, nonlinear, and multifactorial nature of financial markets (Brock, Lakonishok & LeBaron, 1992).

Quantitative analysts have long utilized systematic trading strategies and algorithmic models to exploit historical price patterns and behavioral regularities (Chan, 2013). Empirical studies suggest that financial markets, while exhibiting a high degree of noise, are not entirely random and may contain exploitable temporal or structural dependencies (Fama, 1965). This has given rise to numerous predictive frameworks attempting to model such dynamics.

With the rise of big data and advances in computational power, deep learning models have become a popular tool for capturing complex relationships in stock price movements. These models can leverage large volumes of historical financial data to learn non-linear representations, and some have been extended to incorporate exogenous information such as financial news, social media sentiment, and macroeconomic reports (Bollen, Mao & Zeng, 2011). While these heterogeneous data sources provide valuable contextual signals, they also introduce significant noise and inconsistencies, often masking the underlying causal factors that drive asset prices (Tetlock, 2007).

Moreover, financial markets are subject to abrupt structural shifts, regime changes, and unforeseen external shocks such as economic crises or geopolitical events which challenge the generalization capacity of most predictive models. The non-stationarity and high volatility of stock data further complicate both short-term and long-term forecasting tasks.

1.3 Concepts in Stock Prediction

As financial markets evolve in complexity and data availability, stock price prediction has remained a central challenge for both academic researchers and industry practitioners (Cont, 2001). Over the decades, various analytical paradigms have emerged to interpret and forecast market behavior (Lo, 2004). These range from classical theories grounded in fundamental and technical analysis (Stott, Alsac & Monticelli, 1987) to statistical models that formalize time-series dependencies. More recently, the rise of data-driven methodologies has further expanded the predictive toolkit through machine learning techniques (MATTHEW, 2021).

This section provides a comprehensive overview of the foundational approaches in stock prediction, encompassing traditional fundamental and technical analysis, statistical modeling techniques, and machine learning-based methods. By examining their theoretical underpinnings, methodological frameworks, and practical implications (W. Chen, Hussain, Cauteruccio & Zhang, 2024), we highlight both the strengths and limitations of each paradigm in capturing the complexities of financial time series. While traditional methods offer interpretability and domain-aligned logic, they often struggle to model nonlinear relationships and adapt to the dynamic nature of modern financial systems (Tsay, 2005).

These inherent challenges have catalyzed the emergence of deep learning as a promising direction in stock prediction research. By leveraging hierarchical feature extraction and temporal modeling capabilities (Heckmann, Domont, Joublin & Goerick, 2011), deep learning techniques have demonstrated significant potential in capturing latent patterns and long-range dependencies (Goodfellow, Bengio & Courville, 2016), thereby offering enhanced predictive performance in complex and noisy financial environments (Kelly & Xiu, 2023).

1.3.1 Traditional Analysis

As foundational pillars of financial market analysis, *fundamental analysis* and *technical analysis* have dominated stock prediction paradigms for more than a century (Stott et al., 1987). These classical approaches offer contrasting yet complementary perspectives on market behavior, grounded in distinct theoretical assumptions and analytical methodologies.

Fundamental Analysis

Fundamental analysis is predicated on the belief that a company's intrinsic value can be objectively estimated through a comprehensive evaluation of its financial and economic environment (Abarbanell & Bushee, 1998). This approach typically encompasses three hierarchical levels of analysis:

- **Micro-level factors:** Internal financial metrics such as balance sheets, income statements, cash flow statements, corporate governance structures, and competitive advantages (Light, 2009).
- **Meso-level factors:** Industry-specific dynamics including market share, sectoral growth trends, and competitive landscape (Saviotti & Pyka, 2008).

- **Macro-level factors:** Broader economic indicators such as interest rates, inflation, fiscal policy, and geopolitical developments (Ali, 2020).

This methodology is often associated with long-term investment strategies aimed at identifying discrepancies between a stock's market price and its estimated intrinsic value. It has its roots in Benjamin Graham's value investing framework, which emphasizes a margin of safety and prioritizes fundamental strength over short-term market fluctuations.

Technical Analysis

In contrast, technical analysis disregards a company's financial fundamentals and instead focuses exclusively on market-generated data such as price and volume. It is grounded in the principles of Dow Theory (Rhea, 1993), which posits that:

- Market prices reflect all available information.
- Price movements exhibit persistent trends.
- Historical price patterns tend to recur over time.

Technical analysis transforms historical market data into predictive insights through:

- **Chart pattern recognition:** e.g., head-and-shoulders, double tops/bottoms
- **Technical indicators:** e.g., Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), Bollinger Bands
- **Volume and sentiment analysis:** including momentum oscillators and market breadth indicators

Practitioners of technical analysis typically operate within short- to medium-term horizons, leveraging quantitative tools to detect trend formations and potential reversal points. While rooted in classical charting principles, contemporary implementations increasingly incorporate machine learning and algorithmic trading techniques to enhance pattern recognition and decision-making capabilities.

1.3.2 Traditional Statistical Approaches

Traditional statistical approaches have long served as the foundation of quantitative finance, predating the advent of machine learning techniques by several decades. Dominant from the 1970s to the early 2000s, these models offered mathematically grounded frameworks for understanding and forecasting market behavior (J. D. Hamilton, 2020; Campbell, Lo, MacKinlay & Whitelaw, 1998). Despite being constrained by the computational capabilities of their time, they introduced essential paradigms for time-series modeling in financial contexts.

Core Methodologies

- **Linear Regression:** One of the earliest and most widely adopted models, linear regression sought to explain stock returns based on macroeconomic variables (e.g., interest rates, GDP growth) or firm-specific indicators (e.g., P/E ratios, dividend yields). While interpretable and computationally efficient, its linearity assumption limited its capacity to model complex market dynamics (Menike, Dunusinghe & Ranasinghe, 2015).

- **ARIMA (AutoRegressive Integrated Moving Average):** Widely regarded as a benchmark for univariate time series forecasting, ARIMA decomposes financial series into autoregressive, differencing, and moving average components. However, the requirement for stationarity often conflicted with real-world financial data, particularly in volatile markets (Box & Pierce, 1970).

- **GARCH (Generalized Autoregressive Conditional Heteroskedasticity):** Designed to model time-varying volatility, GARCH effectively captured volatility clustering observed in asset returns. Its primary utility lies in risk modeling rather than directional return forecasting, and it remains limited in capturing nonlinearity in return dynamics (Bollerslev, 1986).

Inherent Limitations

Despite their historical significance, traditional statistical models exhibit several intrinsic limitations:

- **Strong Assumptive Foundations:** These models often rely on strict statistical assumptions such as stationarity, linearity, and normally distributed residuals assumptions frequently violated in real financial markets, which display fat-tailed distributions, regime shifts, and structural breaks.
- **Manual Feature Engineering:** Model specification, including lag selection and parameter tuning, typically requires extensive domain expertise. This reliance on human intervention introduces subjectivity and reduces scalability.
- **Limited Adaptability:** Static in nature, these models struggle to accommodate evolving market conditions, such as the emergence of high-frequency trading and rapid structural changes in financial systems.

Although these methods demonstrated reasonable performance under stable market conditions (Lo, Mamaysky & Wang, 2000), their inability to capture nonlinear relationships and latent variables such as investor sentiment and geopolitical risk has led to their gradual replacement by machine learning and deep learning-based approaches in modern stock prediction systems.

1.3.3 Machine Learning-Based Methods

With the rise of data-driven paradigms in the 2010s, machine learning (ML)-based approaches emerged as a dominant force in quantitative finance (Hastie, 2009). These methods offered significant advancements over traditional statistical models by relaxing restrictive assumptions and enabling the modeling of complex, nonlinear patterns from heterogeneous data sources.

Core Algorithms

Several supervised learning algorithms have demonstrated strong predictive capabilities in financial applications:

- **Support Vector Machines (SVM):** Based on Vapnik's statistical learning theory, SVM constructs optimal hyperplanes in high-dimensional spaces to classify market states. Its kernel function enables nonlinear classification, making it suitable for trend detection and sentiment-based categorization (Hearst, Dumais, Osuna, Platt & Scholkopf, 1998).
- **Support Vector Regression (SVR):** An extension of SVM tailored for continuous output prediction. SVR leverages the ε-insensitive loss function to estimate asset prices and forecast volatility under uncertain conditions (Smola & Schölkopf, 2004).
- **Random Forests:** As an ensemble learning technique utilizing bootstrap aggregation of decision trees, Random Forests provide robust performance in handling missing values and variable importance assessment, particularly effective in processing fundamental and macroeconomic indicators (Breiman, 2001).

Methodological Advantages

Machine learning methods introduced several transformative capabilities to stock prediction:

- **Nonlinear Modeling:** ML algorithms can capture complex, nonlinear interactions among financial variables, such as those between macroeconomic indicators and asset-specific behaviors (Bao, Yue & Rao, 2017; J. Patel, Shah, Thakkar & Kotecha, 2015).
- **High-Dimensional Data Processing:** ML techniques are adept at processing diverse feature sets, including structured numerical indicators and unstructured sources such as financial news or alternative data (S. I. Lee & Yoo, 2019; Fischer & Krauss, 2018).
- **Adaptive Learning Mechanisms:** Traditional ML approaches can incorporate embedded feature selection via regularization (e.g., Lasso, Ridge) or wrapper methods like recursive feature elimination (Henrique, Sobreiro & Kimura, 2019).

Emerging Limitations

Despite their practicality and success in structured data analysis, classical machine learning (ML) models such as Support Vector Machines (SVM) and Random Forests exhibit several inherent limitations when applied to financial time-series forecasting:

- **Lack of Temporal Dependency Modeling:** Most ML methods assume independently and identically distributed (i.i.d.) samples and cannot capture long-term dependencies across time steps. For example, they fail to model how a positive earnings report from a week ago might affect stock prices today (Fischer & Krauss, 2018).

- **Inability to Represent Structural Interactions:** These models are not designed to handle the graph-based relationships that exist in financial markets, such as industry affiliations, supply chain dependencies, or co-movement patterns between stocks (F. Feng et al., 2019).

- **Limited Multimodal Learning Capability:** Classical ML models struggle to process diverse data modalities such as candlestick chart images, financial news text, and technical indicators in an integrated and scalable manner (Sezer, Gudelek & Ozbayoglu, 2020).

Although classical ML models are often efficient and interpretable, their inability to model sequential, structural, and multimodal relationships has led researchers to increasingly adopt deep learning techniques for stock prediction tasks (Bao et al., 2017).

1.3.4 Deep Learning-Based Methods

As the state-of-the-art paradigm in modern quantitative finance, deep learning-based methods have revolutionized stock prediction through automated feature learning and multi-modal data integration. These approaches overcome traditional limitations by jointly modeling numerical sequences, textual sentiments, and relational structures through hierarchical representations (Sezer et al., 2020). The typical deep learning-based forecasting workflow involves four key stages:

- **Multi-source data acquisition:** Collecting structured (e.g., historical prices, technical indicators) and unstructured data (e.g., financial news, social media, knowledge graphs).

- **Unified feature encoding:** Converting diverse inputs into a numerical format suitable for neural networks, including time-series normalization, word embeddings, and graph representations.

- **Neural architecture processing:** Leveraging deep learning models to extract hidden representations and temporal or structural dependencies from encoded data.

- **Predictive signal generation:** Outputting probability scores, directional predictions, or regression-based forecasts for use in algorithmic trading strategies.

Several specialized deep learning architectures have been widely applied in stock prediction tasks:

- **Recurrent Neural Networks (RNNs):** Designed for sequential data, RNNs maintain a hidden state that evolves over time, making them suitable for modeling temporal dependencies in financial series. However, standard RNNs often suffer from vanishing gradients when dealing with long sequences (Medsker, Jain et al., 2001) .

- **Long Short-Term Memory (LSTM) networks:** An advanced form of RNN, LSTM incorporates gating mechanisms that preserve long-term memory and alleviate gradient issues (Hochreiter & Schmidhuber, 1997). LSTMs have shown strong performance in forecasting stock prices and volatility over multiple time horizons.

- **Convolutional Neural Networks (CNNs):** Originally developed for image recognition tasks, CNNs have been successfully adapted for financial time-series analysis by capturing local temporal patterns through one-dimensional convolutions. They are especially useful in extracting trend signals, technical chart patterns (e.g., candlestick formations), and price momentum from raw time-series or transformed visual representations such as candlestick images (Tsantekidis et al., 2017). CNNs are also commonly used in hybrid architectures, often paired with LSTMs, to jointly learn spatial and temporal dependencies.

By combining these models or incorporating them into hybrid frameworks, researchers can exploit temporal patterns, latent structures, and linguistic cues, thereby enabling more accurate and robust stock market forecasting systems.

1.3.5 Graph Learning-Based Methods

Graph learning-based methods have gained significant traction in financial modeling due to their ability to represent complex relational structures among financial entities. Unlike traditional time-series models that treat stocks as independent units, graph-based models explicitly capture dependencies such as industrial sectors, price co-movements, and news-driven correlations by modeling stocks as nodes and relationships as edges (Z. Wu et al., 2020; Zhao et al., 2022).

Graph Neural Networks (GNNs) serve as the cornerstone of this approach. They use message passing mechanisms to iteratively aggregate information from a node's neighbors, thereby enabling each stock's representation to be enriched with relational context (Kipf & Welling, 2016). Variants like Graph Attention Networks (GAT) further improve performance by dynamically weighting neighbor contributions (Velickovic et al., 2017). Temporal extensions such as Temporal GCNs (F. Feng et al., 2019) incorporate time-sensitive edge weights or node features to model stock behavior more accurately.

Despite their power, most graph learning methods rely on fixed or knowledge-based graph construction (e.g., sector-based or correlation-based edges), which limits adaptability to dynamic market conditions. Recent studies aim to learn the graph structure dynamically from data, or augment static graphs with semantic embeddings from textual data.

1.3.6 Graph Motifs

Graph motifs are small, recurrent subgraph patterns that capture higher-order structural dependencies in graphs (Benson, Gleich & Leskovec, 2016). In the context of financial networks, such as stock co-movement graphs, motifs can represent frequently occurring interaction templates among assets such as leader-follower dynamics, triangular correlations, or feedback loops. Compared to edge-level modeling, motif-based representations provide a more expressive characterization of latent relationships between stocks.

Motif-based Graph Convolutional Networks (MGCNs) enhance traditional GCNs by leveraging motif-induced neighborhoods rather than simple 1-hop edge connections (Monti, Otness & Bronstein, 2018). This allows models to capture higher-order semantics and richer topological features. For example, a triangle motif might represent a cluster of technology stocks that tend to respond similarly to market signals due to industry proximity or shared news sentiment.

Recent advancements extend this idea further by integrating attention mechanisms into motif-based architectures, such as the Motif-based Graph Attention Network (MGSR) (Y. Zhang et al., 2023). MGSR learns attention weights across multiple motif structures, thereby redistributing neighbor importance and mitigating the common over-smoothing problem seen in deep GCNs. This motif-aware attention allows each node to selectively aggregate information from motif-induced neighborhoods, leading to more discriminative embeddings and improved task performance.

However, identifying meaningful motifs in financial graphs remains computationally expensive and often requires domain knowledge. Moreover, integrating motif-induced structures into end-to-end learning systems presents challenges in scalability and generalization. To address these issues, ongoing research explores automatic motif discovery (X. Chen et al., 2023) and lightweight attention mechanisms to improve interpretability and efficiency.

These limitations highlight the need for motif-aware, attention-based graph architectures to enhance the modeling capacity in stock market prediction.

1.3.7 Transformer and LLM

Transformer-based models and Large Language Models (LLMs) have revolutionized both sequential modeling and natural language understanding, making them particularly well-suited to stock prediction tasks involving multimodal inputs.

Transformers (Vaswani et al., 2017) employ self-attention to model long-range dependencies in sequences without relying on recurrence. Their parallelizability and scalability make them ideal for processing high-frequency stock time-series data. Variants such as Informer (Zhou et al., 2021) and Linear Transformer (Katharopoulos, Vyas, Pappas & Fleuret, 2020) have addressed the quadratic complexity issue, enabling efficient modeling of long historical windows in financial prediction tasks.

LLMs like BERT, GPT, and FinBERT (Araci, 2019) provide rich semantic embeddings from financial texts such as news, analyst reports, and LLM-generated summaries. These embeddings can be incorporated into price prediction models as auxiliary features or used to construct semantic similarity graphs, enabling deeper context-aware reasoning. Moreover, LLMs can be prompted to act as virtual financial analysts, generating interpretable summaries that enhance model transparency and trustworthiness (Nie et al., 2024).

The fusion of Transformer models for temporal modeling and LLMs for contextual representation forms the basis of recent state-of-the-art hybrid systems. These models not only improve predictive accuracy but also offer enhanced interpretability through text-based rationales and attention mechanisms.

1.4 Research Questions

Accurate stock market prediction remains a challenging task due to the inherent complexity, volatility, and dynamic nature of financial systems. Traditional statistical models and early machine learning approaches often fall short in capturing the nonlinear dependencies, structural relationships, and semantic context inherent in financial data. Although deep learning models such as Transformers, Graph Neural Networks (GNNs), and Large Language Models (LLMs) have shown remarkable capabilities in modeling time-series data and extracting semantic insights from unstructured text, several key challenges remain unaddressed.

The primary limitations can be summarized as follows:

- **(i) Inability to capture dependencies:** Conventional models fail to effectively capture long-range dependencies and complex topological relationships among stocks.
- **(ii) Insufficient multimodal integration:** There is an insufficient integration of multimodal information, such as technical indicators, price charts, and textual financial commentary.
- **(iii) Lack of latent relationship discovery:** Existing methods lack mechanisms to uncover latent inter-stock relationships from textual signals (e.g., financial news, social media, or analyst reports)—relationships that could be modeled using motif-based graph structures.
- **(iv) Limited interpretability:** The limited interpretability of deep learning-based predictions poses significant concerns for decision-making in high-stakes financial environments.

To clearly define the scope of the three proposed frameworks, we distinguish among three related yet different prediction tasks.

- **Stock trend prediction** focuses on forecasting the overall directional tendency (e.g., upward, downward, or stable) over a medium- to long-term horizon, emphasizing robust directionality rather than short-term noise.

- **Stock price prediction** aims to estimate future numerical values (e.g., price or return) and is typically formulated as a regression problem, requiring fine-grained modeling of temporal variation and multimodal signals.

- **Stock movement prediction** is commonly formulated as a short-horizon classification task (e.g., up vs. down) between consecutive time steps, and is often used for short-term decision-making where fast directional signals are required.

These tasks differ in their prediction targets (direction vs. value), temporal horizons (medium/long vs. short), and output formulations (multi-class direction, regression, and binary classification.

To address these challenges, this thesis proposes three distinct yet complementary modeling frameworks that leverage multimodal data fusion and relational modeling. This dissertation aims to investigate how the integration of advanced deep learning architectures can enhance the predictive performance of stock trend and price forecasting systems. Specifically, the study addresses the following research questions:

- **RQ1:** *How can time-dependent and stock-dependent relationships in the financial market be jointly modeled using Transformer and Graph Neural Network (GNN) architectures, and how can motif-based graph structures be leveraged to capture high-order patterns among stocks and financial news for enhanced trend prediction?*

- **RQ2:** *How can stock price prediction be improved by effectively integrating heterogeneous data modalities including textual financial analyses, time-series indicators, and visual representations such as candlestick charts into a unified* *forecasting framework? In particular, how can the semantic, temporal, and spatial characteristics of these diverse inputs be jointly leveraged to enhance model robustness and generalization across varying market conditions?*

- **RQ3:** *With the growing availability of deployable Large Language Models (LLMs) capable of generating financial analyses from raw market data, how can we construct dynamic inter-stock graphs based on semantic embeddings of these texts? How does the incorporation of such semantically informed graphs influence the performance and robustness of stock movement prediction compared to models without graph-based relational modeling?*

1.5 Contribution

This dissertation makes three major contributions, each aligned with a specific research question and designed to address critical gaps in stock market forecasting. These contributions highlight the rationale behind the proposed frameworks, the technical methodologies adopted, and the improvements achieved in terms of prediction accuracy, data fusion, and graph-based reasoning.

1.5.1 RQ1: Integrating Transformer and GNN with Motif Graph Structures for Stock Trend Prediction

Conventional stock trend prediction methods often treat each stock as an independent unit, failing to account for the intricate and evolving interdependencies that exist among financial assets. However, markets such as the S&P 500 exhibit strong relational behaviors influenced by sector affiliations, co-movement patterns, and the shared impact of financial news (F. Feng et al., 2019). Capturing these relational structures is crucial for improving model expressiveness and forecasting accuracy.

To address this, recent research has turned to Graph Neural Networks (GNNs), which are capable of modeling stock-level interactions by leveraging topological dependencies (Z. Wu et al., 2020). However, conventional GNNs often rely on static or low-order graphs (e.g., industry-based or correlation-based), which may overlook higher-order structural motifs and fail to incorporate temporal signals effectively.

We propose a motif-based framework that jointly leverages Transformer and GNN architectures to overcome these limitations. Specifically, we construct a bipartite graph between stocks and financial news headlines and extract recurring motif structures to reveal latent inter-stock relationships. These motif-based adjacency matrices serve as a rich foundation for a lightweight GCN module, while a Transformer encoder processes temporal features from historical stock price data.

By fusing motif-aware spatial relationships with long-range temporal dependencies, our model effectively captures complex stock dynamics and outperforms traditional sequence-based and shallow graph-based methods. This design addresses the limitations of oversimplified inter-stock modeling and enables more context-aware financial forecasting.

1.5.2 RQ2: Multimodal Integration of LLM, Transformer, and CNN for Stock Price Prediction

Traditional stock price prediction methods tend to rely either on technical indicators derived from numerical market data or on sentiment signals extracted from external sources such as financial news or social media. However, the use of external data introduces considerable noise, stemming from unverified information, biased opinions, and market rumors (A. W. Li & Bastos, 2020; F. Feng et al., 2018). These challenges limit the reliability and generalizability of models that depend heavily on exogenous sources.

To address this, we propose a multimodal deep learning framework that exclusively leverages market-derived information while enhancing it through large language models (LLMs). Instead of using potentially noisy external narratives, we employ prompt-engineered LLMs (specifically ChatGPT-4o) to generate concise and interpretable technical analysis summaries directly from structured historical data. These include derived insights from indicators such as Moving Averages, RSI, MACD, and Bollinger Bands.

The generated summaries are embedded using FinBERT (Araci, 2019) to capture semantic richness. Simultaneously, historical stock data is transformed into visual formats (e.g., candlestick and volume charts) and processed through a Convolutional Neural Network (CNN) to extract spatial dynamics. Temporal dependencies are learned using a Linear Transformer (Choromanski et al., 2020), which offers efficient sequence modeling with reduced computational complexity (Katharopoulos et al., 2020).

By integrating semantic (LLM), spatial (CNN), and temporal (Transformer) features into a unified representation, the model effectively captures diverse patterns in stock movements. Experimental evaluation on the S&P 500 dataset from 2022 to 2023 shows that this architecture outperforms baseline unimodal models, demonstrating the effectiveness of using LLMs as domain-specific analytical agents for financial forecasting.

1.5.3 RQ3: LLM-Augmented Enhanced Graph Transformer for Stock Movement Prediction

Traditional graph-based models for stock prediction often construct static graphs based on industry classifications or historical price correlations (Y. Chen, Wei & Huang, 2018; F. Feng et al., 2018). However, such heuristics fail to capture dynamic semantic relationships and evolving inter-stock dependencies shaped by market sentiment and news events. Moreover, most prior approaches treat textual information as auxiliary input, rather than integrating it structurally into the graph modeling process.

To address these limitations, we propose an LLM-Augmented Enhanced Graph Transformer (LLM-GT) framework that unifies numerical stock data and semantically-informed graph structures derived from textual financial analysis. Our method is motivated by the increasing availability of locally deployable LLMs that can act as domain-specific financial analysts capable of interpreting market indicators and generating concise daily commentary.

In our implementation, we select 260 representative stocks from the S&P 500 index and deploy the DeepSeek-R1-Distill-Qwen-14B model to generate daily 20-word financial summaries based solely on technical indicators (e.g., MACD, RSI, Bollinger Bands). These summaries are transformed into semantic embeddings using FinBERT (Araci, 2019), which is pretrained on financial corpora and excels at capturing nuanced contextual sentiment.

A static semantic graph is constructed based on the cosine similarity of FinBERT embeddings aggregated over time. Each node represents a stock, and an edge is established between nodes whose semantic similarity exceeds a fixed threshold. This strategy avoids relying on noisy external sources (e.g., public news or social media) while still capturing latent semantic relationships encoded in LLM-generated financial rationales.

We then integrate this graph with a Graph Transformer architecture (Dwivedi & Bresson, 2020), combining both numerical time-series features and FinBERT-based semantic vectors. The model applies adjacency-aware multi-head attention to learn meaningful representations of each stock, capturing both structural relationships and temporal dynamics.

Empirical evaluations on the 2024 S&P 500 dataset demonstrate that our LLM-GT model significantly outperforms traditional GNNs (GCN, GAT) and sequence models

(LSTM, Transformer), validating the effectiveness of combining semantic reasoning from LLMs with graph-based relational learning.

1.6 Thesis Outline

This thesis is organized into six chapters, each addressing a specific aspect of the research problem. The overall structure is designed to build a coherent narrative from foundational knowledge to novel contributions, culminating in the evaluation and conclusion of the proposed frameworks.

- **Chapter 2: Literature Review**
  This chapter provides a comprehensive survey of existing methods in stock market prediction. It reviews classical approaches including fundamental and technical analysis, traditional statistical models, and the evolution toward machine learning, deep learning, and graph-based approaches. Recent advances in Transformer architectures and Large Language Models (LLMs) are also discussed, identifying critical research gaps that motivate this thesis.

- **Chapter 3: Motif-Based Graph and Transformer Integration for Stock Trend Prediction**
  Addressing **RQ1**, this chapter presents a novel hybrid model that integrates motif-based graph structures with Transformer and GNN architectures to enhance stock trend prediction. The chapter details the graph construction method, motif selection strategies, and the temporal encoding using Transformer models. Extensive experiments on the S&P 500 dataset validate the effectiveness of this approach.

- **Chapter 4: LLM-Enhanced Multimodal Deep Learning Framework for Stock Price Prediction**

In response to **RQ2**, this chapter introduces a multimodal framework that integrates LLM-generated financial text, visual time-series data, and temporal sequences. It describes how an LLM (ChatGPT-4o) is used to simulate a virtual analyst, followed by the use of FinBERT, CNNs, and a Linear Transformer for multimodal feature fusion. The chapter demonstrates that the proposed integration significantly improves prediction performance.

- **Chapter 5: LLM-Augmented Enhanced Graph Transformer for Stock Movement Forecasting**

  This chapter tackles **RQ3** by proposing an LLM-Augmented Enhanced Graph Transformer model. The method generates daily textual summaries using LLMs, which are semantically embedded via FinBERT to construct a context-aware graph. A Graph Transformer is then employed to capture both structural and semantic dependencies among stocks. The chapter presents implementation details, graph design choices, and a thorough experimental evaluation.

- **Chapter 6: Conclusion**

  This chapter provides a comprehensive summary of the thesis contributions and identifies several open challenges that point to promising avenues for future research.

Chapter 2

Literature Review

Following the introductory chapter, which outlined the motivation, research objectives, and overarching structure of this thesis, the current chapter provides a comprehensive review of existing literature in the field of stock market prediction. The aim is to contextualize the proposed methodologies by examining prior work across several key domains, including classical statistical approaches, machine learning models, deep learning architectures (e.g., LSTM, CNN, Transformer), and recent developments in graph neural networks (GNNs) and large language models (LLMs). This literature review establishes the theoretical foundations and identifies the gaps and limitations in current methods that this thesis aims to address. By doing so, it sets the stage for the three manuscripts that follow, each of which introduces a novel framework integrating advanced neural architectures and semantic representations for enhanced stock market forecasting.

2.1 Introduction

This chapter presents a comprehensive literature review of stock prediction methodologies, structured into four major categories: traditional linear models, machine learning approaches, deep learning models, and graph learning-based models. The review examines the strengths and limitations of each method, highlighting the evolution from statistical to deep and relational approaches in financial forecasting.

2.2 Traditional Methods for Stock Market Prediction

2.2.1 Linear Models

Linear models have long served as baseline approaches in stock market prediction tasks, particularly due to their simplicity and computational efficiency. Two key advantages of linear models are: (1) they can be built and trained quickly, and (2) they are capable of processing large volumes of time series data efficiently.

The Autoregressive Integrated Moving Average (ARIMA) model generalizes the Autoregressive Moving Average (ARMA) model by incorporating three components: Autoregression (AR), Integration (I), and Moving Average (MA). ARIMA has demonstrated strong performance in traditional time series forecasting tasks (Ariyo, Adewumi & Ayo, 2014). However, its effectiveness in financial applications is limited by several underlying assumptions. Specifically, ARIMA assumes the input time series is weakly stationary and follows a linear and normally distributed process. In contrast, real-world stock market data often exhibit strong nonlinearities, high noise, and nonstationary behavior, making ARIMA-based models less suitable for such environments.

Another notable linear model is the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, which extends the Autoregressive Conditional Heteroskedasticity (ARCH) framework. While ARCH models describe time-varying volatility by modeling error variance as a function of previous error magnitudes, GARCH improves upon this by modeling the error variance using an ARMA structure. This allows GARCH to capture both short- and long-term volatility clustering in financial time series (Bollerslev, 1986).

Jie Gao (J. Gao, 2023) proposed a combined ARIMA-GARCH model for forecasting Chinese stock prices. The hybrid model achieved an average relative error of 1.29% under stable market conditions, demonstrating promising short-term performance. However, its performance significantly deteriorated when applied to data from volatile markets, due to its inability to model nonlinearities and structural breaks effectively.

In summary, while linear models such as ARIMA and GARCH are foundational in time series forecasting, they struggle to capture the complex, nonlinear patterns commonly found in stock market data. Their reliance on stationarity and linearity limits their adaptability to real-world financial dynamics.

2.2.2 Machine Learning Models for Stock Prediction

Machine learning (ML) models are widely used for stock prediction due to their ability to learn complex relationships between input features and target labels. These models are typically trained on historical data under the assumption that the training and testing datasets share a consistent distribution in the feature space. Once trained, the models can be applied to unseen data to predict future stock trends.

One key advantage of ML models is their capability for automatic feature extraction and dimensionality reduction, which helps improve predictive accuracy, particularly in high-dimensional time series contexts. Commonly used ML algorithms in stock market forecasting include Linear Regression (LR), Random Forest (RF), and Support Vector Machines (SVM).

Trafalis and Ince (Trafalis & Ince, 2000) applied SVM to predict short-term stock price trends and compared it with the performance of Multilayer Perceptron (MLP) and ARIMA models. Their results showed that SVM outperformed both. Wang, Li, and Bao (S. Wang, Li & Bao, 2017) proposed a Novel Advanced-Fuzzy Support Vector

Machine (NA-FSVM) model that enhances traditional SVM by integrating fuzzy logic, and applied it to predict the direction of stock price trends using NASDAQ and S&P market data, reporting improved predictive performance compared to conventional SVM approaches. In contrast, Linear Regression has been shown to outperform Support Vector Machine models in certain stock price prediction settings, particularly when the underlying relationships are relatively simple and approximately linear. Moreover, stock prediction is often formulated as a classification task that focuses on predicting the directional movement of prices rather than their exact values. Under this formulation, ensemble-based methods such as Random Forest have demonstrated strong performance in capturing stock price trends.

While ML models provide powerful nonlinear mapping capabilities that compensate for the limitations of traditional time series methods, they still face challenges in capturing long-range temporal dependencies inherent in stock data (Qiu, Wu, Ding, Xu & Feng, 2016).

2.3 Deep Learning Methods for Stock Prediction

Deep learning (DL) algorithms have become increasingly prominent in stock market prediction due to their capacity to model complex, nonlinear relationships and extract meaningful patterns from high-dimensional data. Compared with traditional machine learning methods, deep learning models can automatically learn hierarchical features through multi-layer architectures, enabling both temporal and spatial dependencies to be captured effectively (Schmidhuber, 2015).

Among the most widely used deep learning architectures are Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and more recently, Transformer-based models. These models have demonstrated superior performance in time series forecasting, price movement classification, and sentiment-based prediction tasks in financial markets.

Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Chung, Gulcehre, Cho & Bengio, 2014), are well-suited for modeling sequential data such as stock price time series. LSTM networks mitigate the vanishing gradient problem through memory cells and gating mechanisms, allowing them to learn long-term dependencies more effectively. However, RNN-based models often struggle to model complex inter-stock relationships and can be computationally expensive for long sequences.

Convolutional Neural Networks (CNNs) are primarily used to capture spatial patterns in data and have been adapted for financial time series by converting historical price data into visual or matrix representations. These models are effective in extracting local trends and technical indicators, especially when combined with temporal models such as LSTMs.

Transformer-based models have recently gained attention for financial forecasting due to their ability to model long-range dependencies without relying on recurrence. The Transformer architecture (Vaswani et al., 2017), originally introduced for Natural Language Processing (NLP), employs self-attention mechanisms that weigh the relevance of each time step dynamically. This enables parallel computation and better scalability. In stock prediction, Transformers have been used to model complex temporal interactions, handle multimodal inputs (e.g., price, news, and social sentiment), and outperform traditional RNN-based models in both regression and classification tasks (Zhou et al., 2021; N. Wu, Green, Ben & O'Banion, 2020).

2.3.1 Recurrent Neural Network Methods for Stock Prediction

Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining hidden states that evolve over time. Each hidden state at time step t incorporates information from the previous state $t-1$, making RNNs naturally suitable for temporal modeling in stock price prediction. However, standard RNNs suffer from the vanishing gradient problem, which limits their ability to capture long-term dependencies (Hochreiter & Schmidhuber, 1997).

To overcome these limitations, Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber (Hochreiter & Schmidhuber, 1997). LSTM enhances the RNN architecture by incorporating gating mechanisms input, forget, and output gates that regulate the flow of information through memory cells. This structure allows LSTM to retain relevant information over extended time periods and mitigate gradient-related issues. As a result, LSTM has become one of the dominant deep learning models for financial time series forecasting.

Several studies have empirically validated the superiority of LSTM over traditional methods. Bao et al. (Bao et al., 2017) proposed a hybrid model integrating wavelet transforms, stacked autoencoders (SAEs), and LSTM (WSAEs-LSTM), achieving significantly improved performance compared to standalone RNN or LSTM models. Nelson et al. (Nelson, Pereira & de Oliveira, 2017) used LSTM to predict stock prices in the Brazilian market and found that it outperformed MLP, Random Forest, and pseudo-random baselines in terms of accuracy and drawdown metrics.

To further enhance LSTM's learning capacity, Zhang et al. (X. Zhang et al., 2019) proposed an attention-based LSTM (AT-LSTM) model for financial time series prediction, where an attention mechanism is used to assign adaptive weights to input feature sequences before feeding them into the LSTM. This approach enhances the ability of the model to focus on the most relevant temporal features and improves prediction performance compared to traditional LSTM baselines. Similarly, Kulshreshtha et al. (Kulshreshtha et al., 2020) proposed a hybrid ARIMA–LSTM model for stock market prediction using live data, where ARIMA is used to model the linear structure of the time series and LSTM captures nonlinear patterns. Their results indicated that the hybrid approach can outperform using ARIMA or LSTM alone in their experimental setting.

Fischer et al. (Fischer & Krauss, 2018) explored stacked and bidirectional LSTM models for predicting the S&P 500 index, and demonstrated their effectiveness over classical machine learning methods such as logistic regression and random forest. However, they also noted that LSTM models may struggle to identify the sources of systemic financial risk.

Siami-Namini et al. (Siami-Namini, Tavakoli & Namin, 2019) conducted a comparative study between ARIMA and LSTM on various financial time series. Their results showed that LSTM consistently reduced the Root Mean Square Error (RMSE) by 84%–87% relative to ARIMA.

Several representative RNN-based models have been proposed for stock prediction, integrating a variety of data sources and network structures:

- **Bao et al. (2017)** proposed a deep learning framework combining wavelet transforms, stacked autoencoders (SAEs), and Long Short-Term Memory (LSTM) networks. The model was tested on six major indices: CSI200, Nifty50, Hang Seng, Nikkei 225, S&P500, and DJIA. Their approach applied wavelet transforms for denoising, SAEs for feature extraction, and LSTM for regression. Results showed that this WSAEs-LSTM model outperformed both standalone RNN and LSTM models (Bao et al., 2017).

- **Sawhney et al. (2020)** developed a Multipronged Attention Network (MAN) to predict stock movements using price sequences, social media texts, and inter-company relationships. GRU encoded the price data, while textual information from the StockNet dataset was embedded using GloVe and the Universal Sentence Encoder. The combined features were passed through a Graph Attention Network (GAT), with additional company relations extracted from Wikidata (Sawhney, Agarwal, Wadhwa & Shah, 2020).

- **Xu and Cohen (2018)** introduced StockNet, which includes a Market Information Encoder (MIE) using GRUs, a Variational Movement Decoder (VMD), and an Attentive Temporal Auxiliary (ATA) module. Their model integrates both price and textual information and was trained on two years of stock data for 88 companies (Y. Xu & Cohen, 2018).

- **Chen (2021)** proposed the Fine-Tuned Contextual Embedded Recurrent Neural Network (FT-CE-RNN), which uses contextual embeddings from a fine-tuned BERT model to process financial news headlines. Unlike other models, this approach does not include any stock price data (Q. Chen, 2021).

- **Kim and Kim (2019)** developed a feature fusion LSTM-CNN model for stock price forecasting, using historical S&P 500 ETF data. CNNs were applied to image representations of stock charts, while LSTM handled numerical time series. The fused features led to significantly reduced prediction errors (T. Kim & Kim, 2019).

These findings underscore the effectiveness of LSTM in capturing complex temporal dependencies, making it a cornerstone technique in stock trend forecasting.

2.3.2 Convolutional Neural Network Methods for Stock Prediction

Convolutional Neural Networks (CNNs) have demonstrated strong capabilities in spatial feature extraction and are particularly effective when input data can be represented in image-like structures. Initially introduced for image recognition (Krizhevsky, Sutskever & Hinton, 2012), CNNs have since been adapted for financial applications, such as identifying technical indicators including moving averages and Bollinger

Band (Bollinger, 1992). In stock market prediction, CNNs are often employed after transforming time series data into matrix or visual formats.

Sezer et al. (Sezer & Ozbayoglu, 2018) proposed a novel approach that converts daily stock data into 15×15 grayscale images, where rows represent technical indicators and columns represent time intervals. Each image is labeled as a trading signal buy, sell, or hold depending on subsequent price movement. A CNN is then trained to classify these signals. Their evaluation demonstrated the practical effectiveness of applying image-based techniques to financial time series forecasting.

Hoseinzade and Haratizadeh (Hoseinzade & Haratizadeh, 2019) developed CNNpred, a CNN-based framework supporting both 2D and 3D CNN architectures. The 2D-CNNpred model is used for single-market forecasting, while the 3D-CNNpred model generalizes across multiple markets. Their framework integrates a diverse set of input features, including macroeconomic and technical variables. Although Principal Component Analysis (PCA) was used for dimensionality reduction, the authors noted that this may have limited the model's ability to filter out noise effectively, impacting overall accuracy.

Kim and Kim (T. Kim & Kim, 2019) introduced a feature fusion model that combines CNN and LSTM networks to jointly learn from temporal and spatial representations of financial data. In their design, historical stock prices are fed into LSTM for sequence modeling, while corresponding image-based features are input into CNN for parallel training. The fused model significantly reduced forecasting errors compared to models using only one type of representation.

Several studies have explored the application of CNN-based architectures for stock price prediction by transforming financial data into image-like or grid-based formats.

Singh and Srivastava (Singh & Srivastava, 2017) employed Deep Neural Networks (DNN) and Radial Basis Function Neural Networks (RBFNN) to predict Google's stock price using technical indicators. They transformed the data into a two-dimensional matrix of size 36×20, where 36 represents technical features and 20 is the time window. The model predicted the 21st-day price based on the past 20 days. Their results demonstrated that the DNN achieved 15.6% and 43.4% higher accuracy compared to RBFNN and RNN, respectively. However, the study did not consider inter-stock relationships or external information.

Bao, Yue, and Rao (Bao et al., 2017) proposed a hybrid deep learning framework combining wavelet transforms, stacked autoencoders (SAEs), and LSTM for financial time series forecasting. The model was tested on six major stock indices: CSI200, Nifty50, Hang Seng, Nikkei 225, S&P500, and DJIA. It used daily price data, technical indicators, and macroeconomic variables. The wavelet transform was used to reduce noise, SAEs for feature extraction, and LSTM for final regression. Their results showed that the WSAEs-LSTM model significantly outperformed standalone RNN or LSTM models.

Sezer and Ozbayoglu (Sezer & Ozbayoglu, 2018) converted time series stock data into 15×15 grayscale images based on technical indicators and their respective time intervals. They labeled each image with a trading signal (buy, sell, or hold) and trained a CNN to classify these signals. The study was among the first to convert stock data into image form and demonstrated the feasibility of CNNs in predicting trading actions. However, their model did not incorporate external signals such as financial news or sentiment.

Hoseinzade and Haratizadeh (Hoseinzade & Haratizadeh, 2019) introduced CNNpred, a CNN-based prediction framework with two variants: 2D-CNNpred for individual markets and 3D-CNNpred for modeling inter-market dependencies. Their approach used a diverse set of input features and applied Principal Component Analysis (PCA) for dimensionality reduction. While 3D-CNNpred successfully modeled cross-market relationships, the use of PCA was found to be suboptimal in removing data noise, slightly affecting prediction accuracy.

These studies collectively show that CNNs, especially when combined with feature engineering or temporal models, can offer powerful tools for learning stock patterns. However, the choice of input representation and the lack of sequential modeling remain key limitations. Despite their strengths, CNNs exhibit some limitations when applied to time series forecasting. First, CNNs typically require fixed-size input, making them less suitable for variable-length sequences common in financial data. Second, CNNs are not inherently designed to capture temporal dependencies or inter-stock relationships, which may restrict their performance in complex market environments.

2.3.3 Transformer-based Methods for Stock Prediction

Transformer-based architectures have recently gained significant attention in the domain of stock market prediction due to their capability to model long-range temporal dependencies and their inherent support for parallel computation. Originally designed for sequence-to-sequence tasks in natural language processing (Vaswani et al., 2017), Transformers have been adapted for time series forecasting and financial modeling through various enhancements.

Unlike recurrent models, Transformers do not rely on sequential processing, which enables better scalability and efficiency in training. The self-attention mechanism within Transformers dynamically captures the importance of each time step, making them particularly suitable for modeling the temporal dependencies present in stock market data.

Several Transformer variants have been applied to stock prediction tasks. Zhou et al. (Zhou et al., 2021) introduced Informer, which leverages ProbSparse self-attention to improve time series forecasting efficiency for long sequences. Similarly, Wu et al. (N. Wu et al., 2020) proposed the Deep Transformer for Time Series Forecasting (DTFT), demonstrating its superiority over RNN-based models in financial prediction.

In addition, researchers have explored combining Transformers with other modalities and architectures to enhance predictive performance. For example, Zong et al. (Zong, Wan, Cascone & Zhou, 2025) proposed a multimodal stable fusion framework via a gated cross-attention mechanism to integrate numerical indicators, textual documents, and relational graphs for stock movement prediction. These multimodal approaches reflect the growing importance of integrating diverse data sources (e.g., news, tweets, macroeconomic indicators) for market analysis.

Some studies have also integrated Transformer modules with Graph Neural Networks to leverage both temporal and relational dependencies. Feng et al. (R. Feng, Jiang, Liang & Xia, 2025) proposed Temporal Graph Attention Network (TGAT), which introduces time encoding into the attention mechanism for modeling dynamic interactions over time. Such models offer promising directions for capturing both the structure and evolution of financial markets.

Transformer-based models have shown strong potential in stock prediction tasks due to their expressive capacity, scalability, and ability to integrate multimodal data. Their integration with GNNs represents a frontier area for future financial forecasting systems.

2.4 Graph Neural Network Methods for Stock Prediction within a Classification Framework

Graph Convolutional Networks (GCN) (Kipf & Welling, 2016) enable relational reasoning, improving performance by incorporating stock relationships.

To use Graph Neural Networks (GNN) (Scarselli, Gori, Tsoi, Hagenbuchner & Monfardini, 2008) for stock prediction, one can represent stocks as nodes to facilitate node classification tasks. This leverages the stock relationships to enhance prediction

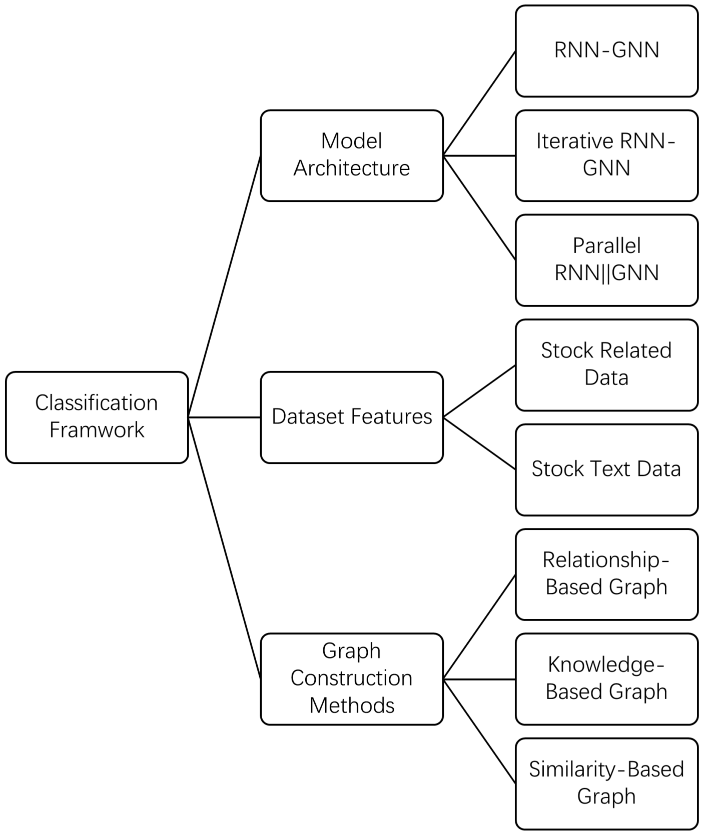


Figure 2.1: Classification Framework.

models. Chen et al. (Y. Chen et al., 2018) demonstrated GNN's potential in stock market representation and prediction.

we propose a novel classification framework that analyses existing approaches from three aspects, including *model architecture, dataset feature and graph construction*. Figure 4.1 illustrates the proposed taxonomy based on the classification framework.

2.4.1 Model Architecture

We have identified three types of model architectures that can represent most existing literature: (1) RNN-GNN architecture, (2) iterative RNN-GNN architecture, and (3) parallel RNN-GNN architecture. Figure 2.2 shows the structure of the three architectures. The RNN-GNN architecture is the most commonly used for graph-based stock

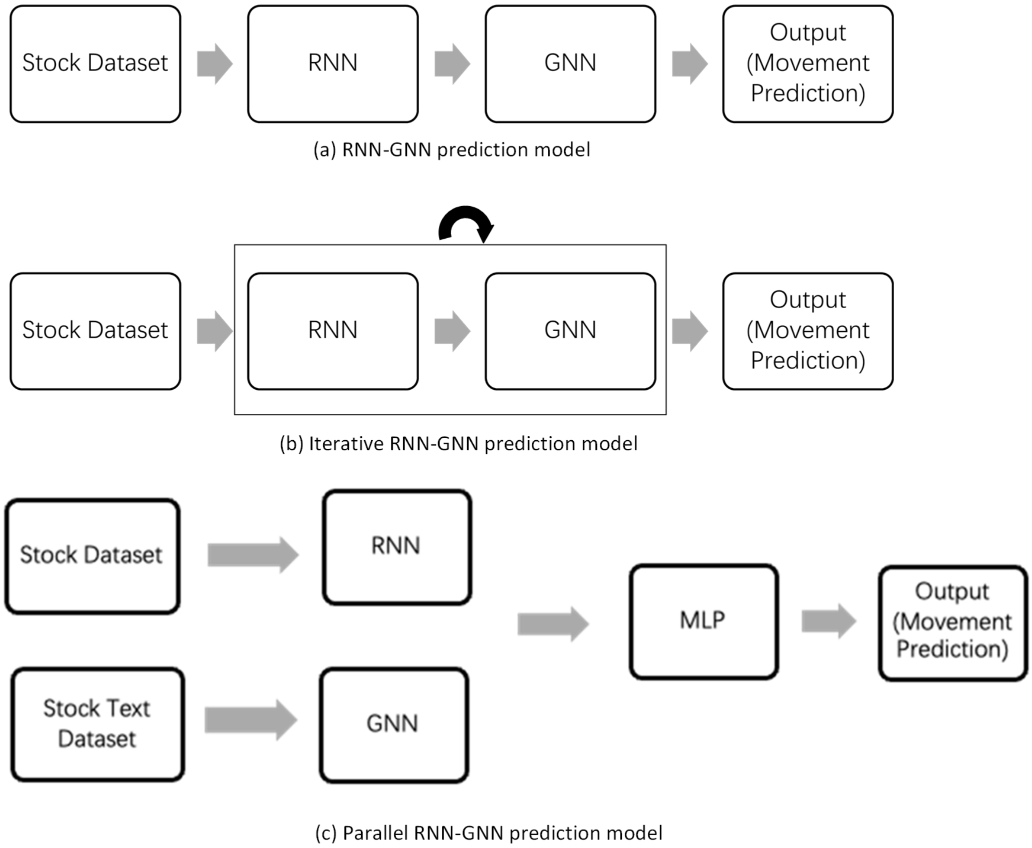


Figure 2.2: Common architecture using GNN for stock prediction.

market prediction. Stock time series data are first fed into a Recurrent Neural Network, such as LSTM (Hochreiter & Schmidhuber, 1997), GRU (Chung et al., 2014), or Bi-directional (Lafortune & Willems, 1993), to extract features. Then, the hidden features are processed by a Graph Neural Network, such as GCN (Kipf & Welling, 2016), Graph Attention Network (GAT) (Velickovic et al., 2017), or GraphSAGE (W. Hamilton, Ying & Leskovec, 2017), for node classification. The other two architectures are less common in the literature.

As shown in Figure 2.2 (b), the iterative RNN-GNN architecture achieves deeper integration and collaboration between the RNN and GNN, including an information exchange mechanism, resulting in better capture of the complex relationships between temporal and relational data.

The parallel RNN || GNN architecture, shown in Figure 2.2 (c), effectively captures both temporal and relational information in complex datasets. This integration facilitates a more comprehensive understanding of the data, leading to improved performance in tasks such as sequence modeling, time series forecasting, and relational reasoning.

Compared with the other two architectures, the RNN-GNN model is simpler and easier to implement, as it uses the output of the RNN directly as input to the GNN. However, it may underperform in some complex tasks. The choice of architecture depends on task-specific requirements and data characteristics.

2.4.2 Datasets Features

Stock price fluctuations depend on numeric and text data, including prices and volumes (Vijh, Chandola, Tikkiwal & Kumar, 2020), as well as announcements and social media (F. Li et al., 2010). Using historical prices helps predict future trends, with numeric data being more standardized and accessible. Text data, requiring sentiment analysis for integration (J. Li & Meesad, 2016), contributes to forecasting but cannot solely predict prices. Combining both data types enhances prediction accuracy.

2.4.3 Graph Construction Method

In literature on GNN for stock market prediction, three primary methods for graph construction are identified: correlation-based, knowledge-based, and similarity-based graph constructions.

Relationship-based graph construction (Y. Chen et al., 2018) constructs graphs from internal stock relationships, utilizing key market indicators such as prices and volumes to create nodes for each stock. Historical data analysis, through correlation coefficients or time series models, determines edges reflecting stock correlations.

Knowledge-based graph construction(Y. Liu, Zeng, Ordieres Meré & Yang, 2019)(Zhao et al., 2022) leverages external information, such as sector dependencies or economic impacts, to enhance graph structure. This approach integrates expert insights or industry reports, capturing relationships beyond dataset information.

Similarity-based graph construction (W. Li et al., 2021) builds graphs by identifying similarities among stocks, useful for implicit relationship mapping. Nodes represent stocks, with edges based on similarity scores (e.g., cosine similarity or Euclidean distance), highlighting clusters of similar stocks.

2.5 Comparative Analysis of GNN-based Models

In this section, we review existing approaches that use GNNs for the stock market. We discuss each approach from three aspects: architecture, dataset features, and graph construction methods, as proposed in the previous section.

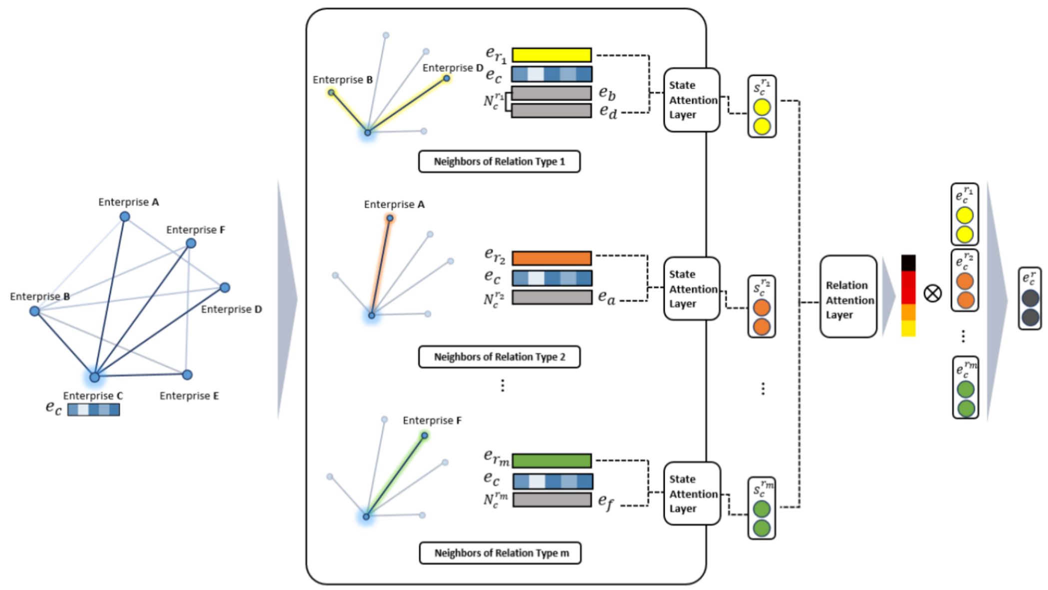


Figure 2.3: Hierarchical Attention Network for Stock Prediction.

2.5.1 Model Architecture

1. RNN-GNN architecture

Most models adopt the RNN-GNN architecture, embedding time series data for graph-based prediction. Combining RNNs' ability to capture temporal sequences with GNNs' insight into stock interrelations, data is encoded using LSTM or GRU for pattern recognition and long-term dependencies.

Once the time series data is embedded, it is fed into the GNN component. The GNN utilizes graph convolutional layers to propagate information among the interconnected stocks in a graph structure. By considering the correlations and dependencies between stocks, the GNN can capture collective behaviour and market dynamics.

In (Y. Chen et al., 2018), Chen et al. propose a joint RNN-GNN model called the Incorporating Corporation Relationship-Graph Convolutional Neural Network (ICR-GCN), which employs the RNN-GNN architecture. The model is composed of two parts. The first part is an LSTM that encodes time series information to extract features for each company. These features then serve as the node attributes in a graph that represents the relationships between companies. Subsequently, a three-layer GCN is applied for node classification.

$$Y = \mathrm{softmax}\left(\widehat{A}ReLU\left(\widehat{A}ReLU\left(\widehat{A}X'W\right)W\right)W\right)$$

where $\widehat{A}$ is the adjacency matrix, X' represents the historical features, W is the learnable weight matrix. $ReLU(\cdot)$ and $softmax$ functions are used as the activation functions, and cross-entropy is used as the loss function.

Compared to similar studies, the work in (Sawhney et al., 2020) distinguishes itself by considering the impact of social media text on stock prices. By integrating social media text with financial data and stock relationships, the study introduces additional dimensions of signals for stock prediction.

In (R. Kim et al., 2019), Kim et al. propose a Hierarchical Attention Network for Stock prediction (HATS) that uses relational data for stock market prediction. It selectively aggregates information on different relation types and adds the information to the representations of each company. As shown in Figure 2.3. HATS is the RNN-GNN model; it has three layers including the Feature Extraction layer, the Relational Modeling layer and the Prediction Layer. In the feature extraction layer, one of LSTM and GRU is used to encode features.

The Relational Modeling layer is used to encode the graph structure. The HATS layout is shown in Figure 2.3. e_{r_m} is the relation type, e_n is the feature of node n, and $N_i^{r_m}$ is the set of neighboring nodes of i for relation type m.

The prediction layer focuses on individual stock and S&P500 Index movements, classifying stocks into three categories: up, down, and neutral, using a linear transformation for individual predictions. Mean pooling calculates the index's graph representation. HATS, similar to ICR-GCN, employs RNN-GNN architecture but differs by using GAT, whereas ICR-GCN utilizes GCN.

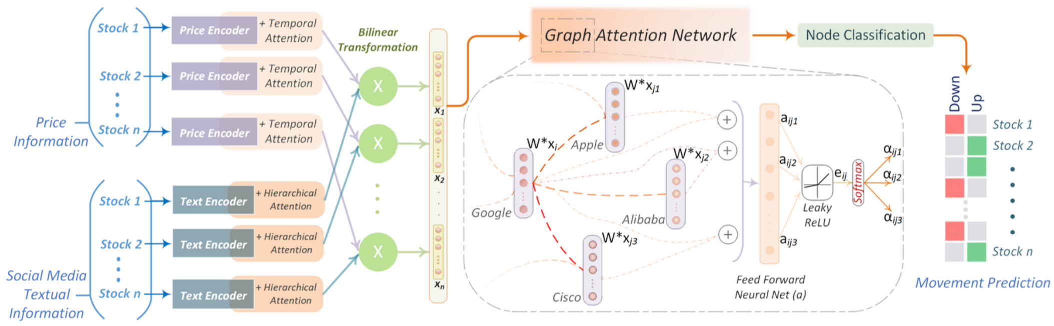


Figure 2.4: MAN-SF Model.

HATS selectively aggregates information from different types of relationships and adds the information to the representation of each company.

In Feng et al.'s study (F. Feng et al., 2019), the Relational Stock Ranking (RSR) framework employs the Temporal Graph Convolution (TGC) model for stock prediction, featuring a three-layered joint RNN-GNN model: a sequential embedding layer with LSTM for capturing stock sequences, a relational embedding layer using TGC for stock interconnections, and a ranking scores prediction layer for stock ranking. Different from other models, RSR-TGC uniquely captures temporal dynamics with TGC, distinguishing it from GCN and GAT models.

This approach contrasts with Wang et al. (H. Wang, Li, Wang & Zheng, 2021), which uses a hierarchical model to analyze temporal stock relationships and market dynamics across multiple timescales, considering financial, social media sentiment, and other factors.

In (Sawhney et al., 2020), Sawhney et al. propose a Multipronged Attention Network for Stock Forecasting (MAN-SF) by learning from historical prices, social media, and inter-stock relations. It is made up of a hierarchical attention network and a GAT. The Hierarchical Attention Network (HAN) is responsible for capturing relevant signals across diverse data, while the GAT is responsible for predicting stock movements.

MAN-SF model is a joint RNN-GNN architecture. As shown in Figure 2.4, first, GRU is used as a Price Encoder (PE) that takes the prices of a stock over a period of time and uses that to produce a price feature. The temporal attention mechanism is a way of aggregating information from different time steps into an overall representation. This is done by assigning learned weights to each time step, which allows the most important information to be aggregated together. For example, the formula of temporal attention mechanism $\zeta(\cdot)$ is shown as:

$$\zeta\left(\bar{h}_z\right) = \sum_i \beta_i h_i \tag{2.1}$$

$$\beta_i = \frac{\exp\left(h_i^T W \bar{h}_z\right)}{\sum_{i=1}^{T} \exp\left(h_i^T W \bar{h}_z\right)} \tag{2.2}$$

where $\bar{h}_z$ is the hidden states of GRU, β_i is the attention weight and W is the learnable parameter matrix.

Secondly, the Social Media Information Encoder (SMI) employs GRU to distill tweet data, using a hierarchical attention mechanism to encode this information into vectors. Thirdly, the Blending Multimodal Information layer merges features from PE and SMI, applying a bilinear transformation for learning price-tweet interactions, optimizing the mix of data inputs. Lastly, stock movement prediction is performed using a GAT.

Similar to the above models, MAN-SF also uses the RNN-GNN architecture pattern; the main difference is that three attention mechanisms are used to extract features, which include price data, news data and stock relations data by temporal attention, hierarchical attention and graph attention.

2. Iterative RNN-GNN architecture

In (), Li et al. propose an LSTM Relational Graph Convolutional Network (LSTM-RGCN) model that predicts the overnight stock movement based on the correlation

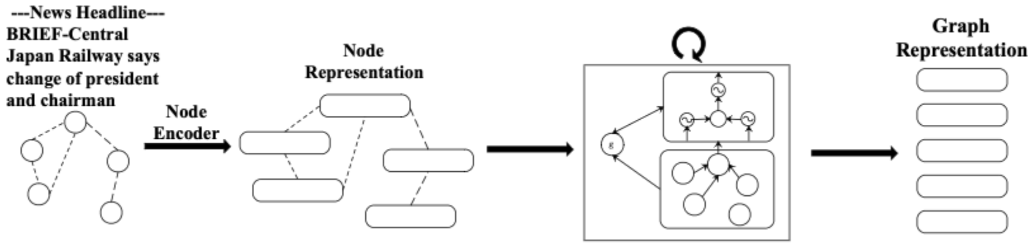


Figure 2.5: LSTM-RGCN Model.

between stocks. This paper constructs a graph by converting each stock's news into a vector and calculating the relationship between each stock by using the cosine similarity. LSTM-RGCN is the first model that unitized connection among stocks to predict the movement of stocks that are not directly associated with news.

The LSTM-RGCN model, depicted in Figure 2.2(b), embodies an iterative RNN-GNN architecture. As illustrated in Figure 2.5, the process begins with LSTM encoding news data into vectors. Subsequently, the model merges the news vector with the node embedding to form the node vector. Finally, RGCN encodes the graph structure.:

$$N^{l+1} = \sigma\left(\sum_r D_r^{-\frac{1}{2}} A_r D_r^{-\frac{1}{2}} H^l W_r^l + W_h H^l\right) \tag{2.3}$$

where A_r is the adjacency matrix of relation r, $D_r^{-\frac{1}{2}} A_r D_r^{-\frac{1}{2}}$ is the normalized adjacency matrix. W_r^l is the learnable parameter matrix. W_h is the learnable parameter matrix for the node vector. The parameter matrices are shared across layers. H^l represents the hidden representations of all the nodes in the l-th layer. N^{l+1} is the aggregated neighbor information for the $(l+1)$-th layer.

Finally, the model predicts stock movement based on the node representation in the graph. $Sigmoid(\cdot)$ and $softmax$ functions are used as the activation functions. The cross-entropy is used as a loss function for this two-class classification task.

3. Parallel RNN || GNN architecture

In their paper (Zhao et al., 2022), Zhao et al. proposed a method called Dual Attention Networks to learn Stock Movement Prediction (DANSMP). This method leverages a market knowledge graph to model the relationships between stocks and make predictions about stock momentum. The graph comprises various types of information, including the relationships between companies and their executives.

DANSMP integrates three layers: stock sequential embedding, stock relational embedding, and prediction.

Initially, it merges technical and sentiment features using GRU for feature extraction. The relational layer uses dual attention networks for spillover signal representation, focusing on company-executive relationships via inter- and intra-class networks. Inter-class networks compare company and executive features, while intra-class networks assess same-type entity interactions.

Finally, embeddings are combined in a neural network for stock movement prediction. DANSMP's innovation lies in its parallel RNN-GNN structure and dual attention mechanism, enhancing market relationship analysis.

2.5.2 Dataset Feature

1. Data Information

The ICR-GCN dataset, sourced from Tushare API, comprises CSI 300 historical prices for listed companies from 29/04/2017 to 31/12/2017. It features five numeric attributes per company: open, close, high, low, and volume, utilizing seven days of historical data for input.

HATS dataset uses S&P 500 historical price data of listed companies between 08/02/2013 and 17/06/2019 (in total, 1174 trading days). For each company, there are three numeric features including open price, close price, and volume. The authors use

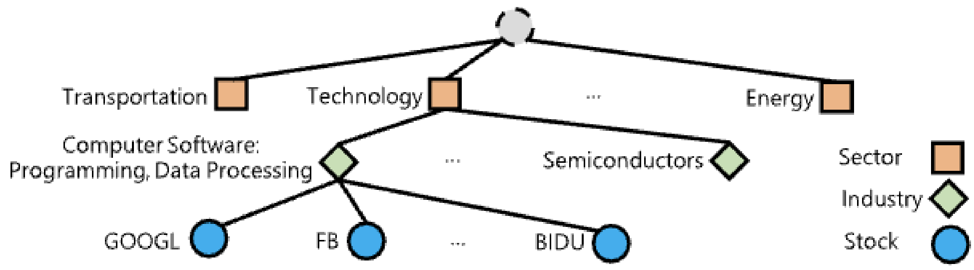


Figure 2.6: Sector-industry relations and wiki company-based relations.

historical price change rate $\boldsymbol{R}_i^t = \frac{(P_i^t - P_i^{t-1})}{P_i^{t-1}}$ as model input, where P_i^t is the closing price at time t of a company i and P_i^{t-1} is the closing price at time $t-1$.

In the RSR-TGC model, data collection comprises three categories, starting with sequential price data from New York Stock Exchange (NYSE) and NASDAQ between February 1, 2013, and December 8, 2017. This dataset encompasses open, close, high, low, and volume metrics, using twenty-nine daily close prices for calculating (Moving Average) MA5, MA10, MA20, and MA30. These indicators, combined with the closing price, undergo normalization for model input.

The MAN-SF dataset used in this study was the StockNet dataset (Y. Xu & Cohen, 2018) which contains data on high-trade-volume stocks in the S&P 500 index in the NYSE and NASDAQ markets. The dataset is split into three parts: training, validation, and testing. The training data was used to train the MAN-SF model.

The DANSMP includes total of 185 stocks from the China Securities Index 300(CSI300E) and 73 stocks from the CSI100E are used to create two datasets which are collected from the China Securities Index (CSI). The market dataset includes historical price information (opening price, close price, highest price, lowest price and volume).

2. Stock Text Information

The second type in RSR-TGC model is sector-industry relations and the third type is Wiki relations such as supplier-consumer relations and ownership relations.

The LSTM-RGCN model also includes Financial news and market data from Tokyo Stock Exchange (TSE) from 1/1/2013 to 29/08/2018 from Reuters. There are two numeric features including open price and close price that are used to calculate the overnight movement. The formula for calculating the overnight movement is shown below:

$$\text{Movement} = \left(p_o^t - p_c^{t-1}\right) / p_c^{t-1} \tag{2.4}$$

where $p_o^t$is the open price of the current trading day and p_c^{t-1} is the close price of the previous trading day. Global Vectors for Word Representation (GloVe) (Pennington, Socher & Manning, 2014) and Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee & Toutanova, 2019) are used as word embedding on financial news to generate model input features.

In DANSMP model also includes the news dataset from four financial mainstream sites from wind, sina, hexun and sohu.

2.5.3 Graph Construction Method

1. Correlation-based Graph Construction Methods

The ICR-GCN utilizes a data-driven graph construction method based on financial investment facts, using a graph of companies (stocks). This graph construction approach is designed to capture relevant relationships between companies. In this weighted graph, each node corresponds to a company, while the edges connecting the nodes represent the relationships between these companies. Furthermore, the weight assigned to each edge reflects the shareholding ratio between the connected companies.

2. Knowledge-based Graph Construction Methods

The HATS model constructs a heterogeneous graph from Wikidata to analyze relations between entities such as companies and persons. It simplifies this graph into a company-focused homogeneous one using a meta-path, representing companies as nodes connected by various relationship types, such as 'Owned by'.

The RSR-TGC model constructs its graph using sector-industry information, as illustrated in Figure 2.6, and company relations sourced from Wikidata. It categorizes stocks by industry and establishes connections between stocks based on first and second-order relations derived from Wikidata (Vrandečić & Krötzsch, 2014).

MAN-SF leverages first and second-order relations from Wikidata to map the S&P 500 index stocks' relationships, focusing on direct company connections.

DANSMP employs a Bi-typed Hybrid-relational Market Knowledge Graph (MKG) from Tushare API data, featuring company and executive entities with both explicit (directly stated relationships) and implicit (inferred from attributes) relations. This approach enables a detailed analysis of company and executive interconnections for stock movement predictions, enriching market analysis and investment decisions.

3. Similarity-based Graph Construction Methods

In the LSTM-RGCN model, a stock correlation graph is built using historical prices, distinguishing relationships as positive or negative correlations. Connections between stock nodes are made if their similarity's absolute value meets a set threshold, enabling the model to understand stock interdependencies from historical prices.

Table 2.1: Comparison of Models and Metrics Between Articles

Framework	**Model**	**Dataset**	**Graph Relationships**	**Auxiliary Data**
ICR-GCN	LSTM-GCN	CSI300	Stock-Stock	Financial investment data from WIND

Continued on the next page

Framework	Model	Dataset	Graph Relationships	Auxiliary Data
HATS	RNN-GAT	S&P500	Stock-Stock Stock-Owner	Wikidata
TGC	LSTM-TGC	NASDAQ NYSE	Stock-Stock	Wikidata
MAN-SF	GRU-GAT	S&P500	Stock-Stock Stock-Owner	Wikidata
RGCN	LSTM-RGCN	TPX500 TPX100	Stock-Stock	News
DANSMP	GRU-DAN	CSI100E CSI300E	Stock-Stock Stock-Owner Owner-Owner	News

2.6 Discussion and Conclusion

This chapter has reviewed and compared a wide spectrum of stock prediction methodologies, ranging from traditional statistical models to state-of-the-art deep learning and graph-based approaches. A summary of the reviewed GNN-based models is presented in Table 2.1, comparing their frameworks, datasets, graph relationships, and auxiliary data.

From the comparison, it is evident that the RNN-GNN architecture is the most widely adopted framework. Three of the reviewed studies use GCN-based models (ICR-GCN, TGC, RGCN), while the other three leverage GAT-based models (HATS, MAN-SF, DANSMP). In terms of datasets, MAN-SF and HATS share a common dataset based on the S&P 500, whereas the others vary, covering markets in China, the U.S., and Japan. Most models integrate both price and textual information, highlighting the importance of multimodal inputs.

Graph construction strategies also differ notably across studies. Most models adopt homogeneous graphs using stock-to-stock relationships derived from Wikidata or historical data. However, DANSMP stands out by employing a heterogeneous graph that captures complex relations between companies and executives, enabling richer market representation.

2.6.1 Research Gaps and Future Directions

Based on this review, we identify two key opportunities for advancing stock market prediction research:

1. **Integrating Transformers into GNN-based models**: As Transformer architectures excel at modeling long-range dependencies, incorporating them into GNN frameworks may improve predictive accuracy, particularly in modeling long-term temporal and structural dynamics.

2. **Constructing dynamic spatio-temporal graphs**: Most current studies rely on static graph structures. Introducing dynamic graphs that evolve over time can better reflect the real-time changes in market conditions and inter-stock relationships, allowing models to capture the temporal evolution of financial networks.

Additionally, future research may benefit from the following directions:

- **Multimodal learning**: Incorporating and effectively fusing price data, textual sentiment, and macroeconomic indicators remains a challenge. Transformer-based multimodal fusion shows promise in this area.

- **Explainable AI (XAI)** in stock prediction: GNNs and Transformers often act as black boxes. Developing interpretable architectures is critical for adoption in high-stakes financial environments.
- **Reinforcement learning with GNNs**: Emerging work explores combining reinforcement learning with relational modeling for strategy optimization. While not the focus of this review, this presents a valuable future direction.

2.6.2 Conclusion

Stock market prediction is a complex task influenced by numerous dynamic and interrelated factors. This chapter has provided a comprehensive literature review of conventional, machine learning, deep learning, and graph neural network methods. In particular, it introduced a unified classification framework for GNN-based approaches based on model architecture, data types, and graph construction strategies.

The findings underscore the efficacy of graph-based methods in capturing relational dependencies among stocks. With the ongoing evolution of financial markets and increasing data complexity, integrating temporal, structural, and multimodal information through hybrid architectures such as Transformer-GNN models represents a promising research trajectory. This review provides both a foundation and a roadmap for future explorations in intelligent stock forecasting systems.

Chapter 3

A Motif Graph Convolution Network for Stock Prediction

This chapter presents the first core manuscript of this thesis, which investigates the integration of motif-based graph representations with Transformer encoders for stock market prediction. It addresses limitations of static graph structures by identifying recurring subgraph patterns (motifs) that capture higher-order stock correlations. By incorporating these motifs into a motif-aware Graph Convolutional Network (GCN) and combining them with a Transformer for temporal modeling, the proposed model significantly outperforms classical GCN and RNN-based baselines on the S&P 500 dataset. This chapter lays the theoretical and architectural foundation for using graph structures in financial prediction and sets the stage for the subsequent exploration of multimodal and language-enhanced models.

3.1 Overview

The prediction of stock market price trends has always been a challenging issue, attracting widespread attention from both economists and computer scientists. Recently, integrating stock prices with news data has been shown to be an effective strategy for enhancing the accuracy of the prediction task. Yet, many current methods fail to fully leverage the intricate inter-stock relationships inherent in stock news. Applying deep learning, especially Graph Convolutional Networks (GCNs), to predict stock trends has demonstrated advanced performance. This method employs a message-passing architecture, enabling nodes to progressively aggregate information from neighboring nodes across multiple layers. In this research, we propose a novel approach: the Motif-based Graph Convolutional Network for Stock Prediction (MGCN-SP). This model mitigates the over-smoothing problem by incorporating network motifs into the layer propagation process. Specifically, we first generate a motif graph by correlating stocks with stock news. Then, we encode the stock price information and stock news into features using the scaled dot product attention adapted from the transformer architecture. After that, we apply the motif-based graph convolutional network. This framework is designed to jointly refine the embeddings of both stock news and stock time series data using a transformer encoder and to estimate the likelihood of target movements. Finally, we conducted extensive implementation in the U.S. stock market, and the experimental results demonstrate that our method outperforms some state-of-the-art approaches.

Stock market prediction is a highly complex and dynamic task, with various key factors affecting price movement. Capturing and reflecting these dominant factors is crucial. In addition to commonly known factors, unexpected and rare market events can introduce new influential factors. Due to the inherent high volatility of stock markets, prediction accuracy and precision are of paramount importance in this research field.

Recently, deep learning technologies have been extensively applied and researched in stock price trend prediction (Q. Li, Tan, Wang & Chen, 2020). Scholars are exploring the integration of a broader spectrum of data as inputs into models to enhance predictive accuracy. This data encompasses various technical indicators (Merello, Ratto, Oneto & Cambria, 2019), market factors (F. Feng et al., 2018), financial reports (Ballings, Van den Poel, Hespeels & Gryp, 2015), online news articles (A. W. Li & Bastos, 2020), and social media content (H. Wang et al., 2021), among others.

However, traditional time series models often treat financial data as independent and identically distributed, which does not align with the actual dynamics of financial markets (Awartani & Corradi, 2005). For instance, stocks within the same sector typically exhibit stronger correlations compared to those across diverse sectors (Y. Chen et al., 2018). Consequently, recent research has shifted towards utilizing graphs to encapsulate the intricate network of relationships among financial entities (D. Cheng, Yang, Wang, Zhang & Zhang, 2020). This includes the application of graph-based learning techniques for forecasting stock price movements (Q. Liu, Cheng, Su & Zhu, 2018), which have been proven effective by incorporating stock correlations into their predictive frameworks (R. Cheng & Li, 2021). Employing graph-based strategies offers the potential to derive more nuanced and meaningful representations of the input data, thereby enhancing prediction accuracy (D. Cheng et al., 2019).

Despite advancements in graph convolutional network (GCN) applications for stock prediction (J. M.-T. Wu, Li, Herencsar, Vo & Lin, 2023; Long, Chen, He, Wu & Ren, 2020; W. Li et al., 2021), a critical issue known as over-smoothing persists. This issue occurs when node embeddings become disproportionately influenced by a few high-degree nodes after several layers (K. Xu et al., 2018). Our study addresses this challenge by integrating network motifs, which are recurring, significant subgraph patterns found more frequently in complex networks than in randomized ones (Milo et al., 2002). This approach enriches layer-wise propagation in GCNs, allowing nodes to become more attuned to their local structural context and redistributing the weights of neighboring nodes based on their presence in various motifs.

In the domain of stock prediction, where the complexity and interconnections of data are critical, our approach is specifically tailored to leverage the unique characteristics of bipartite graphs commonly used in this field. Typically, these bipartite graphs are structured with one set of nodes representing stocks and the other encompassing various stock-related information, such as news articles about these stocks. This structure allows for the exploration of relationships between stocks and a diverse array of relevant data points. By harnessing the distinct motifs within these bipartite graphs, our methodology effectively enhances the accuracy of stock market predictions, capitalizing on the intricate web of connections between stocks and their related informational elements.

In this work, We introduce a novel approach for creating motif adjacency matrices corresponding to each identified motif. Through these matrices, we develop a motif-oriented graph attention mechanism, designed to integrate complex, higher-order data encapsulated within the motifs. Subsequently, we present the Motif-based Graph Convolutional Network for Stock Prediction (MGCN-SP), a new model conceptualized for forecasting stock market trends. Our proposed model integrates a transformer-based encoder-decoder architecture with a graph convolutional network to efficiently capture and utilize both sequential and structural information for stock prediction.

Our main contributions are summarized as follows:

1. We propose a novel approach to analyse the relationships between stocks and their associated news using motifs by constructing motif adjacency matrices.
2. We propose a framework that combines Transformer and motif-based Graph Convolutional Networks for the task of stock trend prediction.
3. We collected the S&P 500 dataset from the US stock market and applied our proposed methodology and model for analysis. In comparison with other deep learning methods that do not utilize Motif graphs, our proposed framework demonstrated superior performance, outperforming the alternatives.

The rest of this research is organised as follows. Section 3.2 provides an overview of the work related to our research. In Section 3.3, we define motifs and introduce the motif-based graph convolution self-attention approach, along with a detailed presentation of the MGCN-SP model. Section 3.4 details the experimental setup, comparative results, and in-depth analyses. Finally, Section 3.5 offers concluding remarks and summaries of the research's findings.

3.2 Related work

3.2.1 Machine Learning Models for Stock Prediction

The field of stock market forecasting has significantly evolved, marked by a shift from traditional statistical methods to advanced machine learning techniques (Thakkar & Chaudhari, 2021). Initially, stock prediction primarily relied on linear models such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (Awartani & Corradi, 2005) and Autoregressive Integrated Moving Average (ARIMA) (Ariyo et al., 2014). Despite their statistical rigor, these models were limited by the volatile and unpredictable nature of financial markets. Jie Gao (Q. Li, Dong, Wang, Zhang & Wu, 2021) proposed a combined ARIMA-GARCH model to forecast China's stock prices, achieving an average relative error of 1.29% in short-term predictions with stable market data. However, the model underperformed with data reflecting market fluctuations and various uncertain factors. The primary disadvantage of linear models is their requirement for time-series data to be stable, or made stable through differencing. Acknowledging these limitations, recent research has shifted towards employing machine learning algorithms, which offer more robust and adaptive prediction capabilities.

Machine learning models, including Support Vector Machines (SVM) (Joachims, 1998), decision trees (Cramer, Ford & Hall, 1976), Artificial Neural Networks (ANN) (Agatonovic-Kustrin & Beresford, 2000), and Random Forests (RF) (Breiman, 2001), have demonstrated enhanced performance in capturing the complexities of stock data. These methods have been applied in various configurations to improve the accuracy of stock trend predictions. For instance, Lee et al. (M.-C. Lee, 2009) study successfully integrated SVM with hybrid feature selection techniques for trend prediction. Picasso et al. (Picasso, Merello, Ma, Oneto & Cambria, 2019) proposed a novel combination of RF, SVM, and ANN for stock prediction tasks. Parray et al. (Parray, Khurana, Kumar & Altalbe, 2020) explored the synergistic use of SVM, ANN, and logistic regression to predict next-day stock trends. These advancements highlight the dynamic nature of stock market prediction research and the increasing reliance on machine learning methodologies to navigate its complexities.

3.2.2 Deep Learning Methods for Stock Prediction

Within the current paradigm of predictive analytics for financial markets, there has been a pronounced increase in the utilization of deep learning methodologies. Methods such as Long Short-Term Memory networks (LSTM) (Hochreiter & Schmidhuber, 1997), Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012), Gated Recurrent Units (GRU) (Chung et al., 2014), Graph Neural Networks (GNN) (Scarselli et al., 2008), and attention-based mechanisms such as Transformer (Vaswani et al., 2017) have emerged as frontrunners. Their increasing prominence can be attributed to their efficacy in modeling the intricate, non-linear relationships and spatial-temporal dynamics inherent in stock market data. In (Bao et al., 2017), they combined wavelet transforms, stacked autoencoders, and LSTM to forecast stock prices. In (Sezer & Ozbayoglu, 2018), they introduced a method of converting stock market data into 2-D images, using CNNs for stock trading decisions. The 2-D image size is 15x15, which includes 15 technical indicators over 15 intervals for each day. They transformed stock market data into images, marking the buy and sell points, and labeled the images as buy, sell, or hold. CNNs were then used to determine these points in stock prices. In (J. Liu, Lu & Du, 2019), they presented a model based on a corporate knowledge graph embedding system to extend corporate-related news and combined stock news sentiment with stock market data features using the GRU model to predict stock prices.

3.2.3 Graph Neural Networks for Stock Prediction

Graph-based neural architectures, as delineated in the seminal work by Scarselli et al. (Scarselli et al., 2008), offer a framework for directly interfacing with data embodied in graph formats. Such architectures deploy mechanisms for aggregating and updating nodal representations by leveraging the interconnected information from adjacent nodes and their corresponding edges (Y. Wu, Chen, Yin, Ding & King, 2023). In financial applications, individual stocks are typically conceptualized as nodes within a network, interconnected by edges that signify the existence of inter-stock relationships. The task of forecasting trends across an array of stocks often translates into a node classification challenge, aptly addressed through graph neural network constructs (Khemani, Patil, Kotecha & Tanwar, 2024). The intrinsic strength of GNNs lies in their capacity to assimilate inter-stock relationships, enriching the predictive model with relational data nuances. In the context of stock market representation, Chen et al. (Y. Chen et al., 2018) utilized a graph-structured approach wherein companies are symbolized as nodes, and the interrelations among stocks are articulated through edges. Kim et al. (R. Kim et al., 2019) present a novel hierarchical attention network designed for stock prediction (HATs), employing relational data to forecast stock market movements. This approach distinctively assimilates information across varied relation types, enhancing the individual representations of companies.

3.2.4 Network motifs and high-order Graph Neural Networks

Graphs encapsulate the intricate web of financial markets, with network motifs serving as the critical connectors within this complex structure, embodying high-order connectivity (Milo et al., 2002).

Scholarly investigations have affirmed the impact of high-order connectivity in enhancing graph-based machine learning algorithms (Ahmed et al., 2020; Yang, Liu, Zheng & Han, 2018). One such approach, DeepGL (Rossi, Zhou & Ahmed, 2018), innovatively deduces inductive relational functions by employing motifs. Studies on high-order network representations have confirmed the superiority of motif-infused matrix computations in generating more effective embeddings (Rossi, Ahmed & Koh, 2018).

In the realm of financial forecasting, these high-order graph structures can be pivotal. Hierarchical motif convolution has been put forward as a technique for graph classification, identifying important sub-graph structures (Yang et al., 2018). This concept extends to designing graph convolutional frameworks tailored for heterogeneous networks, harnessing the power of motif-driven connectivity. The research also recognizes the limitations of GCN-based models in capturing complex patterns, suggesting a more nuanced high-order framework for graph classification that could be applied to stock prediction (Morris et al., 2019). These advancements in understanding and leveraging network motifs within graph neural networks hold substantial promise for refining the accuracy of stock market trend analysis and forecasting.

3.3 Materials and methods

In this section, we will delve into the foundational aspects of our Motif Graph-based Stock Trend Prediction model, encompassing the data collection methods, the preprocessing techniques applied to the gathered data, and the methodology for constructing a graph that encapsulates the relationships between stocks. We will provide an in-depth exploration of these key components in the following discussion.

3.3.1 Data representation

For the development and validation of our Motif Graph-based Stock Trend Prediction model, we sourced daily trading data of the S&P 500 index from Yahoo Finance, spanning the years 2018 to 2019. Our selection criteria for the dataset included trade dates, trade volumes, and closing prices, which served as both the input features and the targets for our predictions. The comprehensive dataset for the S&P 500 encompasses several key attributes: stock code, trade date, opening price, closing price, daily highest price, daily lowest price, and trading volume.

Additionally, we gathered sectorial information for all companies listed in the S&P 500 index, alongside the titles of news articles pertaining to each stock within the same timeframe. This holistic approach enabled us to incorporate both quantitative market data and qualitative news sentiment into our model, enriching the predictive analysis.

Our dataset comprises 354 training samples and 143 testing samples, derived from working days within the observed period. Each sample is characterized by three features: the S&P 500 index price, and a sequence representing the price movements of all listed stocks over the preceding 30 days.

Given the temporal nature of stock market data, our approach frames stock prediction as a sequence classification task. Therefore, each input sample x is structured as a matrix with dimensions $T \times d$, where T represents the length of the sequence (30 days in our case) and d signifies the dimensionality of the features for each day. This methodological framework ensures a rigorous analysis of temporal patterns in stock prices, facilitating the generation of accurate and actionable predictions through our proposed model.

Following the collection and initial processing of stock market data, we proceeded to preprocess the related stock news to refine our dataset further. This preprocessing involved filtering news content to ensure relevance to the S&P 500 index constituents. Specifically, we eliminated news articles that did not mention any of the S&P 500 stocks or their respective stock codes. This selection criterion was pivotal in aligning the qualitative news data with the quantitative stock market data, thereby streamlining the integration of both data types into our analysis. The filtered news content is essential for constructing a bipartite graph that maps the intricate relationships between stocks and their news articles.

3.3.2 Problem statement and motif definitions

In this section, we introduce motifs within the stock-news bipartite graph, which represents the complex relationships between stocks and news articles. We begin by outlining the preliminaries, defining the problem statement, and presenting the motif definitions. The seven motifs encompass all possible configurations of connections among four nodes in this bipartite structure, ensuring a comprehensive representation of the intricate dependencies and interactions between stocks and news events.

By leveraging these motifs, our approach captures high-order relationships and mitigates the over-smoothing problem often encountered in deep learning models for graph data. This results in a more accurate and interpretable prediction model. The bipartite graph configuration thus provides a robust framework for our prediction model, enhancing the accuracy and interpretability of stock market predictions. Our model can

Table 3.1: Notation

Symbol	Description
$G = (U, V, E)$	Stock-News bipartite graph
$\mathcal{M} = \{M_t\}$	$\mathcal{T}$ motifs, $t = 1, ..., \mathcal{T}$
$\mathcal{A} = \{A_t\}$	$\mathcal{T}$ motif adjacency matrices, $t = 1, ..., \mathcal{T}$
Q, K, V	Query, Key, and Value matrices in attention mechanism
d_k	Dimension of the key vectors
W_i^Q, W_i^K, W_i^V	Weight matrices for query, key, and value in multi-head attention
W^O	Output weight matrix in multi-head attention
mask	Mask matrix used to zero out attention weights for future positions
h_n	Intermediate hidden states in the Transformer architecture
$H_{\text{BERT-news}}$	Feature vectors from news content processed through BERT
$H_{combined}$	Concatenated feature set from Transformer and BERT outputs
$\widetilde{A}_{mc}$	Motif combination matrix
α_m	Weight coefficient of each motif in the motif combination matrix
$\widetilde{A}_m$	Symmetrically normalized motif adjacency matrix
D_m	Degree matrix of motif adjacency matrix A_m
$h_s^{(\cdot)}, h_n^{(\cdot)}$	Updated hidden features for stocks and stock news at layer $l + 1$
$N_m(s), N_m(n)$	Motif-based neighbors of stock s and news n
c_{ui}	Normalization constant
$h_{\text{stock}}^{(L)}$	Stock representation vector from the final MGCN layer L
$\mathbf{W}_{\text{fnn}}, b_{\text{fnn}}$	Weight matrix and bias vector of the fully connected layer
y_{pred}	Predicted binary output
σ	Sigmoid activation function

then offer valuable insights and recommendations based on the interconnected nature of financial markets.

Table 3.1 lists the mathematical notation used in this research.

Preliminaries

[Bipartite graph(Asratian, Denley & Häggkvist, 1998)] In the domain of forecasting stock market trends, where we represent our datasets through bipartite graphs, a bipartite graph $G = (U, V, E)$ is defined by a division of nodes into two exclusive subsets U and V (i.e., $U \cap V = \varnothing$), ensuring that each edge in E connects a node from U to a node in V, thereby prohibiting edges within the same subset. This structure mandates that nodes within either subset, U or V, cannot be directly connected by an edge.

[Stock Relation Graph (SRG)] A Stock Relation Graph (SRG) is an undirected bipartite graph $G = (U, V, E)$, where U denotes the set of stocks, V denotes the set of news articles, and $E = U \times V$ with $e_{ij} \in E$ representing an edge between U and V. If a news article is related to a stock, then the edge e_{ij} exists and is assigned a value $r_{ij} \geq 0$, otherwise $r_{ij} = 0$. In SRG, each edge represents a reference from a news article to a stock, making it an undirected bipartite graph.

[Motif adjacency matrix] A motif-induced graph's adjacency matrix A, referred to as the motif adjacency matrix, is characterized by its entries (i, j), which denote the count of instances where nodes i and j co-occur within a motif. Within the SRG, we identify a collection of $\mathcal{T}$ motifs $\mathcal{M} = \{M_1, ..., M_t, ..., M_{\mathcal{T}}\}$ (where $\mathcal{T} = 7$), and construct a corresponding set of $\mathcal{T}$ motif adjacency matrices $\mathcal{A} = \{A_1, ..., A_t, ..., A_{\mathcal{T}}\}$. Each A_t represents a motif adjacency matrix associated with motif M_t. Specifically, the entry $(A_t)_{s,n}$ signifies the number of instances of motif type M_t that include both stock s and news article n.

Given a node i and a motif M_t, the motif-based neighbors $\mathcal{N}_i^{M_t}$ of node i are defined as the set of nodes that connect with node i in motif M_t.

Motif Proposition

To balance the importance of local structure and higher-order neighborhood connections, we limit our considerations of motifs to those consisting of no more than four nodes.

Provided an undirected bipartite Stock Relation Graph (SRG) $G = (U, V, E)$, there are seven motifs in total, each containing up to four nodes, and the entire set of motifs is $\{M_1, M_2, M_3, M_4, M_5, M_6, M_7\}$. The seven motifs are shown in Figure 3.1, where the blue circles indicate the stock nodes and the green squares indicate the news nodes.

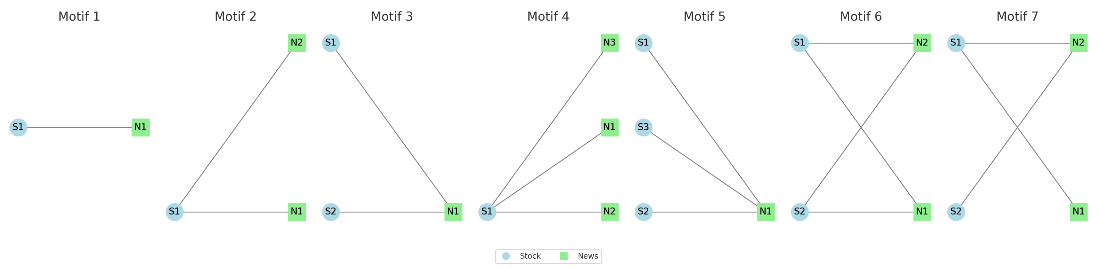


Figure 3.1: Motifs of two to four nodes in undirected bipartite MSRG

Algorithm for Generating Bipartite Motif Adjacency Matrices

In our exploration of stock market prediction through the lens of bipartite graphs, we adopt the algorithms outlined by Yuqi et al. (Y. Zhang et al., 2023) for generating bipartite motif adjacency matrices. This approach first divides edges into groups by one of the two types of nodes, depending on the motif structure, such that in each group the edges have either the same source node or the same destination node. Given these groups, we can then find motif instances and form the motif adjacency matrices. For example, for motif M_2 if ten news nodes are all connected to a stock node, then the ten edges will be in a group and each pair of news nodes in this group will be connected in M_2's adjacency matrix.

Given the bipartite graph structure of datasets in our study, we address the absence of direct connections among same type of nodes by considering the indirect relationships facilitated through the bipartite motifs. This approach significantly influences the feature propagation between stocks and news nodes. As depicted in the following sections, our analysis leverages the Motif-based Graph Convolutional Network (MGCN-SP) to meticulously extract and interpret features from each stock within the bipartite graph. These extracted features are instrumental in enriching the feature set for stock prediction, enabling a nuanced understanding of market dynamics through the lens of the graph's inherent structural patterns.

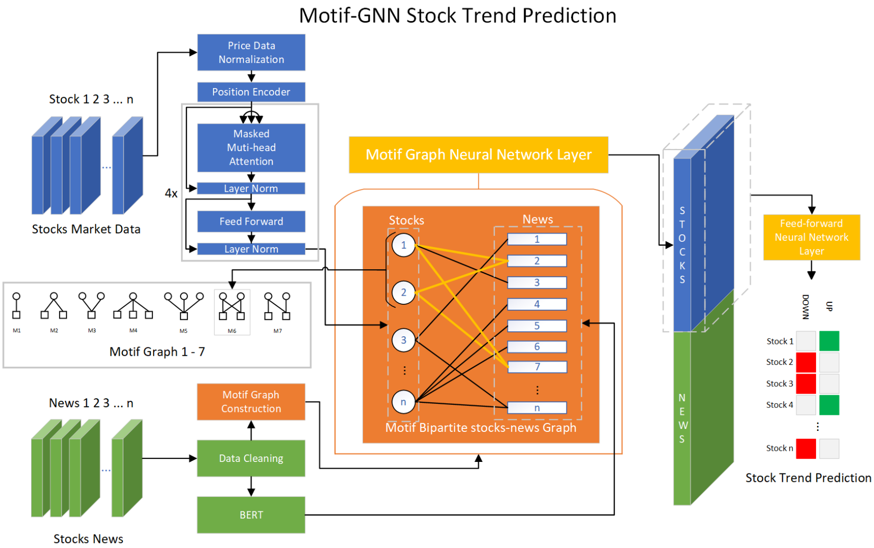


Figure 3.2: MGCN-SP Stock Prediction Framework

3.3.3 MGCN-SP for Stock Prediction Framework

Existing methodologies in the literature (Sawhney et al., 2020) (Zhao et al., 2022) integrate stock price features with text features by directly concatenating these data types, followed by encoding through a unified encoder. This prevalent approach, however, does not fully exploit the latent relationships between stock movements and the information contained within textual news data.

We introduce a comprehensive framework for our proposed MGCN-SP model which is shown in Figure 3.2, detailing its integral components. The model leverages both historical stock prices and stock news as inputs to deduce the probabilities of future stock price movements. To encapsulate the intricate relationships between stocks and associated news, we construct a stock price and stock news bipartite graph. From this graph, we derive motif adjacency matrices, which capture recurrent and significant patterns indicative of stock behavior. Following the construction of motif adjacency matrices, we proceed to process the historical price data and text data from stock news independently. Finally, we integrate the separately encoded features through concatenation, forming a unified representation of each stock's characteristics. This aggregated data is then subjected to MGCN-SP, which is adept at extracting and leveraging the relational information encoded in the motif adjacency matrices. Through this process, our model gains the ability to discern complex patterns and interactions within the stock market, facilitating accurate and robust stock trend predictions.

Feature Extraction and Representation Learning

The Transformer model has significantly advanced sequential data processing, particularly in the domain of stock price prediction. At its core, the self-attention mechanism dynamically evaluates the relevance of each sequence position, enhancing the model's capability to process extended sequences and mitigate issues related to gradient vanishing and explosion:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.1}$$

To capture a diverse array of informational cues, the Transformer employs a multi-head attention strategy:

$$\text{MaskedMultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W^O \tag{3.2}$$

$$head_i = \text{MaskedAttention}(QW_i^Q, KW_i^K, VW_i^V, \text{mask}) \tag{3.3}$$

where the mask is a lower triangular matrix used to zero out the attention weights for future positions.

The architecture is further refined with the integration of residual connections and layer normalization, enhancing training dynamics:

$$h_1 = h_0 + \text{MaskedMultiHead}(h_0, h_0, h_0) \tag{3.4}$$

$$h_2 = \text{LayerNorm}(h_1) \tag{3.5}$$

$$h_3 = h_2 + \text{ReLU}(h_2 W_1 + b_1) W_2 + b_2 \tag{3.6}$$

$$h_4 = \text{LayerNorm}(h_3) \tag{3.7}$$

These Transformer-derived outputs (h_4) are concatenated with vectors from news content processed through the BERT model, creating an enriched feature set ($H_{combined}$). This set serves as input for the MGCN-SP, offering a nuanced examination of market interconnectivity:

$$H_{combined} = \text{Concat}(h_4, H_{\text{BERT-news}}) \tag{3.8}$$

This strategy integrates the analytical strengths of the Transformer and BERT with the MGCN-SP, representing a paradigm shift in financial market analysis to capture a broad spectrum of factors affecting stock prices.

The integration of network motifs enriches feature extraction by leveraging both direct and motif-mediated relationships:

$$H^{(l+1)} = \widetilde{A}_{mc} H_{combined} \tag{3.9}$$

The Motif Graph for stocks and stock news is quantitatively constructed and analyzed using the following two key equations, where m represents different motif types, such as M1 to M7.

The equation(10),

$$\widetilde{A}_{mc} = \sum_{m} \alpha_m \widetilde{A}_m \tag{3.10}$$

defines the motif combination matrix, $\widetilde{A}_{mc}$, as a weighted sum of symmetrically normalized motif adjacency matrices $\widetilde{A}_m$ for each motif type m, weighted by α_m. The coefficient α_m indicates the relative importance or influence of each motif in the combined Motif Graph, allowing for a customized representation of different relational patterns between stocks and stock news.

The equation(11),

$$\widetilde{A}_m = D_m^{-\frac{1}{2}} A_m D_m^{-\frac{1}{2}} \tag{3.11}$$

outlines the normalization process for each motif-specific adjacency matrix, A_m, into its symmetrically normalized form, $\widetilde{A}_m$, using the degree matrix D_m of A_m.

Within the MGCN-SP, the hidden features for stocks ($h_s^{(l+1)}$) and stock-related news ($h_n^{(l+1)}$) are updated based on their motif neighbors:

$$h_s^{(l+1)} = \sum_{m \in M} \sum_{i \in N_m(s)} \frac{1}{c_{ui}} \alpha_m H^l_{combined_i} \tag{3.12}$$

$$h_n^{(l+1)} = \sum_{m \in M} \sum_{u \in N_m(n)} \frac{1}{c_{ui}} \alpha_m H^l_{combined_u} \tag{3.13}$$

Here, $h_s^{(l+1)}$ and $h_n^{(l+1)}$ represent the updated hidden features for stocks and stock news, respectively, at layer $l+1$, incorporating insights from both market history and current news. This methodology highlights a comprehensive approach to understanding the complex dynamics of financial markets.

Prediction Layer

The prediction layer is the apex of the motif-graph architecture, utilizing the refined representations extracted by the MGCN to predict future trends in the stock market. It employs a Fully Connected Neural Network (FNN) for binary classification, distinguishing between potential increases and decreases in stock prices. This distinction is made possible by isolating vector representations specific to stocks from the MGCN's output for processing via the FNN. The methodology for this binary classification is defined as follows:

Given the stock representation vector $h_{\text{stock}}^{(L)}$, derived from the MGCN's final layer L, the prediction layer undertakes binary classification:

$$y_{\text{pred}} = \sigma(\mathbf{W}_{\text{fnn}} h_{\text{stock}}^{(L)} + b_{\text{fnn}}) \tag{3.14}$$

where σ denotes the sigmoid activation function, appropriate for binary classification tasks, with $\mathbf{W}_{\text{fnn}}$ and b_{fnn} representing the fully connected layer's weight matrix and bias vector, respectively. The model's efficacy in classifying stock price trends is evaluated using the cross-entropy loss:

$$\text{Loss} = \frac{1}{N} \sum_i [-y_i \cdot \log(\tilde{y}_i) + (1 - y_i) \cdot \log(1 - \tilde{y}_i)] \tag{3.15}$$

This strategy leverages the MGCN-SP's analytical strength, enabling the prediction of stock price movements by interpreting complex interrelations and patterns, thereby providing a sophisticated method for stock trend prediction.

3.4 Experiments

In this section, we detail our experiment setup and present the results of our evaluation. We include comparisons with baseline models to demonstrate the effectiveness of our MGCN-SP framework. Specifically, we select baseline models from traditional machine learning approaches such as LSTM and GRU, as well as methods used in recent studies, including HATs (R. Kim et al., 2019), AD-GAT (R. Cheng & Li, 2021), and DANSMP (Zhao et al., 2022). This comprehensive comparison underscores the capability of our MGCN-SP methodology to effectively leverage motif information for stock prediction.

3.4.1 Experimental settings

We conduct experiments on the Standard and Poor's 500 dataset (S&P 500). Our dataset spans from January 1, 2018, to the end of 2019, encompassing a total of 500 trading days. The dataset is comprised of 400 training samples and 100 testing samples, corresponding to working days. Each sample encompasses three features: the S&P 500 price and the price sequence of all stocks over the preceding 30 days.

Given that stock prediction can be interpreted as a sequence classification challenge, the input x for each sample is represented by a matrix of dimensions (T, d). Here, T signifies the length of the sequence, set at 30 days for the S&P 500 dataset, and d represents the dimension of the features for each day, which includes the individual stock price features, aggregated cluster features, and S&P 500 index features. Each of these feature vectors is concatenated to form a comprehensive representation of the stock's behavior over the given period.

The labels for each sample are binary, indicating whether the stock price increased or decreased at the end of the prediction window, which is set to 7 days. This approach allows us to evaluate the model's performance in predicting stock price movements based on historical data and derived features.

Hyperparameter Settings

To ensure a fair comparison, we tune the hyperparameters of all the baselines and our model using the same parameters, rather than the original parameters suggested in the papers. The fixed settings for all the models are as follows: the embedding size is set as 240; the batch size is 30.

For our MGCN-SP, the regularization weight λ is set to 1×10^{-4}. The dropout rate is set to 0.6. The feature dimension is set to 50. The number of head attentions is set to 4. We use the Adam optimizer with a learning rate of 0.0001.

3.4.2 Compared Baselines

Our proposed model is rigorously compared with a diverse set of methods across various categories to ensure a comprehensive evaluation.

Classical Time-Series Methods

We include classical time-series methods in our comparison to benchmark the predictive performance of our model against well-established techniques. The methods in this category are:

- **Long Short-Term Memory (LSTM)**: Known for its ability to capture long-term dependencies in sequential data, LSTM is a powerful tool for time-series prediction.
- **Gated Recurrent Unit (GRU)**: Similar to LSTM but with a simpler architecture, GRU is another popular method for handling sequential data.
- **Transformer**: This method has revolutionized sequential data processing with its self-attention mechanism, enabling it to capture long-range dependencies effectively.
- **Informer**: An extension of the Transformer model, Informer is specifically designed for long-sequence time-series forecasting with improved efficiency.

Graph-Based Methods

To further validate our model, we compare it against state-of-the-art graph-based methods that have shown success in stock price prediction:

- **Heterogeneous Attention Networks**:

HATS (R. Kim et al., 2019) uses relational data to selectively aggregate information on different relation types, enhancing stock market prediction.

- **Attribute-Driven Graph Attention Networks**:

  AD-GAT (R. Cheng & Li, 2021) enhances the traditional GAT by incorporating adaptive mechanisms to better capture dynamic relationships.

- **Dual Attention Networks for Stock Movement Prediction**:

  DANSMP (Zhao et al., 2022), leverages a market knowledge graph to model relationships between stocks and predict stock momentum. It integrates stock sequential embedding, stock relational embedding, and prediction layers, using dual attention networks to represent spillover signals.

For a fair and consistent comparison, we adapted the core algorithms of these graph-based methods and integrated them into our framework. Specifically, we ensured that the objective functions of these methods were aligned with our goal of predicting future stock returns. Additionally, we standardized the graph structures used in the comparison by utilizing our Graph and Motif Graph configurations. This approach allows us to isolate the impact of the core algorithms and provides a direct performance comparison across different methodologies.

By benchmarking our model against both classical time-series methods and advanced graph-based approaches, we aim to highlight the robustness and superior predictive capabilities of our proposed MGCN-SP framework.

3.4.3 Evaluation Metrics

In the classification of stock trend prediction, researchers often explore binary outcomes, specifically predicting whether stock prices will rise or fall. The advent of deep learning technology has significantly contributed to advancements in stock trend prediction methods (Riva, Tognollo, Gardumi & Colombo, 2018). Classification metrics play a crucial role in evaluating the effectiveness of these trend prediction models. The relevant formulas are provided below.

Accuracy is defined as the ratio of correctly predicted instances, encompassing both true positives and true negatives, to the total number of cases examined. The formula for accuracy is given as:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}}$$

AUC, or Area Under the ROC Curve, is a metric employed to assess the performance of binary classification models. In the context of stock price movement, where predictions are dichotomized into 'up' or 'down', the ROC (Receiver Operating Characteristic) curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) across different model prediction thresholds. The AUC, which represents the area beneath the ROC curve and ranges from 0.0 to 1.0, serves as an indicator of model accuracy, with higher values denoting superior performance. The formulas for TPR (True Positive Rate) and FPR (False Positive Rate) are as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{TP} + \text{FN}}$$

The AUC is calculated by integrating the TPR over the FPR:

$$\text{AUC} = \int \text{TPR}\, d(\text{FPR})$$

3.4.4 Comparison results

In this section, we first compare the performance of several classical time-series prediction models that exclusively use historical stock data without incorporating Graph Neural Networks (GNN). Next, we evaluate the performance of models that combine time-series prediction techniques with GNNs utilizing non-Motif Graph structures. Finally, we analyze the prediction models using Motif Graphs, and benchmark their performance against the baseline models.

Expanding on this approach, we ensure a comprehensive comparison by employing models such as LSTM, GRU, Transformer, and Informer for the initial time-series predictions. For the subsequent evaluation of GNNs, we integrate these time-series models with different GNN architectures, specifically focusing on the variations in graph structures used. In our final comparison, we rigorously assess how incorporating Motif-based subgraphs enhances the predictive capabilities of our models.

To maintain consistency and fairness in our comparisons, we slightly modify the objective functions of all models to predict future stock returns. This comprehensive approach not only highlights the effectiveness of various prediction models but also provides insights into the potential benefits of integrating motif-based subgraphs in financial forecasting tasks.

From the experimental results shown in Table 3.2, we have the following observations:

GCNs improve time-series models. For all time-series baseline models, their performance are improved by simply integrating a two-layer GCN. This improvement can be attributed to the capability of GCNs to capture the relationships between nodes from their interactions, since time-series models treat all stocks as isolated entities and cannot exploit additional information from their interactions.

Transformer-like architectures remain powerful. Looking at the recent stock

Table 3.2: Performance comparison between out MGCN-SP framework and baseline mathods on S&P 500 dataset.

Method	ACC(%)	AUC	F1-Score
LSTM	56.01	0.5953	0.6517
LSTM+GCN	57.11	0.6087	0.6162
GRU	55.07	0.5631	0.6097
GRU+GCN	56.33	0.5676	0.6162
TRANSFORMER	59.52	0.6520	0.6671
TRANSFORMER+GCN	60.33	0.6399	0.6810
INFORMER	61.65	0.6443	0.6581
INFORMER+GCN	62.23	0.5377	0.7231
HATs	56.47	0.5878	0.6066
AD-GAT	60.63	0.6379	0.6688
DANSMP	61.23	0.5750	0.7141
MGCN-SP	63.38	0.6582	0.7306

prediction models HATs, AD-GAT and DANSMP, they have no obvious advantage over pure Transformer and even underperform pure INFORMER. This is possible because these models cannot efficiently extract and combine information from time-series data and graph data. Improperly adapting GCNs into a large amount of stocks and news data may cause scalability issues.

Comparison with Baseline Models As shown in Figure 3.3:

- **LSTM and GRU Models**: These recurrent models, both standalone and combined with GCN, exhibit lower performance across all metrics. This suggests that they are less effective at capturing the complex dependencies in stock-news data.

- **Transformer-based Models**: Standalone transformer models, as well as those combined with GCNs, show improved performance compared to LSTM and GRU models. However, MGCN-SP still outperforms these methods, indicating the added value of motif-based structures in enhancing model performance.

- **Advanced Models (HATs, AD-GAT, DANSMP)**: These models perform better than the basic recurrent models but still fall short of MGCN-SP's performance.

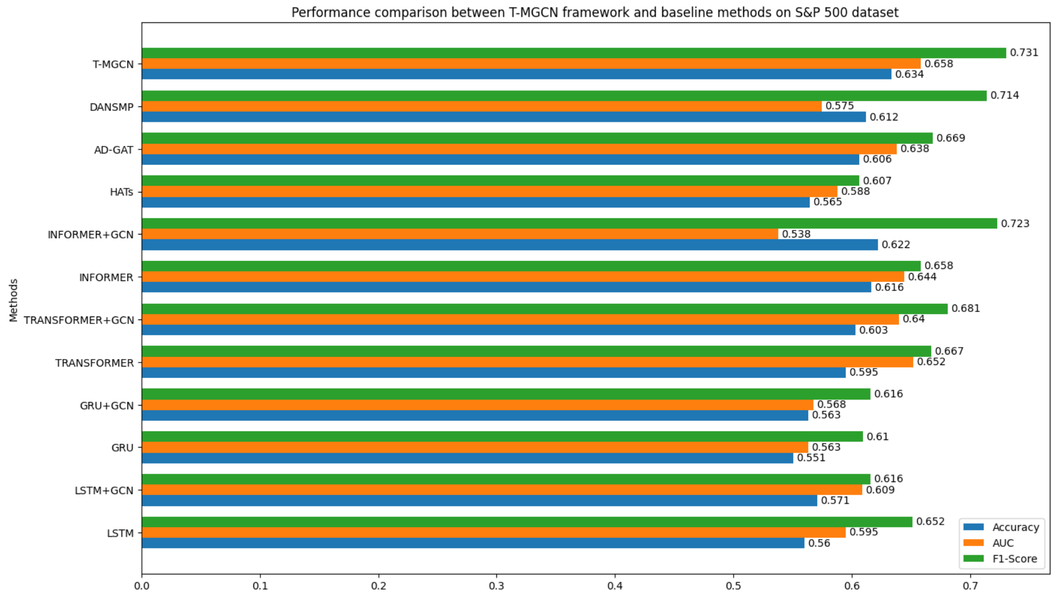


Figure 3.3: Performance Comparison Between MGCN-SP Framework and baseline models

For instance, while AD-GAT achieves a high AUC score of 0.6379, it lags behind MGCN-SP in terms of accuracy and F1-score.

MGCN-SP Outperforms Baseline Models. Our MGCN-SP framework consistently outperforms all baseline models across all evaluation metrics. Compared to Transformer+GCN, which uses two GCN layers, MGCN-SP employs only one MGCN layer yet achieves significantly better performance. This indicates that motifs can efficiently exploit stock-news interactions with fewer layers than GCNs.

3.5 Conclusion

In this chapter, we propose a Motif-based Graph convolutional Network for Stock Trend Prediction. Specifically, we generate motif adjacency matrices from stock-news interactions and then use them for propagation of stocks and news embeddings. Extensive experiments on S&P 500 dataset demonstrate the superior performance of our proposed framework against multiple state-of-the-art models. Through analysis on the experimental results, we can conclude that our motif-based method can enhance stock prediction performance via efficiently capturing the relationships stocks and news. In future work, we plan to incorporate dynamic graph learning into our framework to explore the potential of modeling relational dynamics with considering motif structures.

Chapter 4

LLM-Augmented Linear Transformer-CNN for Enhanced Stock Price Prediction

Building on the graph-based modeling of the previous chapter, this chapter introduces a second manuscript focused on multimodal data fusion and large language model (LLM) integration. While the prior study incorporated graph motifs and structural dependencies, this work expands the input modality to include textual, visual, and numerical features. Specifically, it proposes a hybrid framework that leverages ChatGPT4o to generate technical analysis from market data, transforms these into semantic embeddings via FinBERT, and combines them with features extracted from both a Linear Transformer (for time-series data) and a CNN (for candlestick chart images). This chapter highlights the value of combining LLM-generated insights with deep neural architectures, offering a complementary perspective to graph-based models.

4.1 Overview

Accurately predicting stock prices remains a challenging task due to the volatile and complex nature of financial markets. In this study, we propose a novel hybrid deep learning framework that integrates a large language model (LLM), a Linear Transformer (LT), and a Convolutional Neural Network (CNN) to enhance stock price prediction using solely historical market data. The framework leverages the LLM as a professional financial analyst to perform daily technical analysis. The technical indicators, including moving averages (MAs), relative strength index (RSI), and Bollinger Bands (BBs), are calculated directly from historical stock data. These indicators are then analyzed by the LLM, generating descriptive textual summaries. The textual summaries are further transformed into vector representations using FinBERT, a pre-trained financial language model, to enhance the dataset with contextual insights. The FinBERT embeddings are integrated with features from two additional branches: the Linear Transformer branch, which captures long-term dependencies in time-series stock data through a linearized self-attention mechanism, and the CNN branch, which extracts spatial features from visual representations of stock chart data. The combined features from these three modalities are then processed by a Feedforward Neural Network (FNN) for final stock price prediction. Experimental results on the S&P 500 dataset demonstrate that the proposed framework significantly improves stock prediction accuracy by effectively capturing temporal, spatial, and contextual dependencies in the data. This multimodal approach highlights the importance of integrating advanced technical analysis with deep learning architectures for enhanced financial forecasting.

The stock market, as a vital component of the global financial system, has increasingly attracted investors seeking to capitalize on market trends and maximize their returns. With the constant evolution and expansion of financial markets, the ability to accurately predict stock price movements has become one of the most critical challenges for investors and researchers alike. The stock market is influenced by numerous interconnected factors, including historical prices, trading volumes, economic indicators, global events, and external data such as news and social media. This complexity, coupled with the volatility and uncertainty inherent in financial markets, makes stock price forecasting a highly intricate task.

Accurate predictions are essential for making informed investment decisions. Even small errors in forecasts can lead to significant financial losses, which highlights the need for advanced models that can effectively capture the complex patterns in stock market data. Traditional models that rely solely on time-series data often fall short, as they may not fully account for the various factors influencing stock prices. Stock markets are not simple linear systems; prices fluctuate based on numerous interrelated factors, requiring more sophisticated models to capture these dynamics.

With the rise of deep learning technologies, researchers have increasingly turned to these techniques to enhance prediction accuracy. Deep learning models have the capability to process vast amounts of data and uncover hidden patterns in complex datasets. In recent years, scholars have expanded the scope of input data for stock prediction models to include not only historical prices and volumes but also a broader array of market factors, financial reports, online news articles, and social media content (Q. Li et al., 2020; Merello et al., 2019; H. Wang et al., 2021). These additional data sources offer valuable insights but come with a significant drawback: they introduce a substantial amount of noise into the prediction process, often masking the true underlying patterns and leading to less reliable predictions.

One of the primary challenges in utilizing external data such as news articles or social media content is the presence of noise. Online platforms are flooded with unverified information, rumors, and biased opinions, which can skew predictions when included in stock price forecasting models. While some studies attempt to filter out noise through natural language processing (NLP) techniques (A. W. Li & Bastos, 2020; F. Feng et al., 2018), the task remains difficult, and predictions may still be adversely affected. As a result, there is a growing interest in developing methods that focus exclusively on market data, avoiding the pitfalls associated with noisy external information.

In our study, we address these challenges by focusing on historical market data and leveraging large language models (LLMs) for technical analysis. Specifically, we use prompt engineering with ChatGPT4o to generate high-quality technical indicators from market data, such as moving averages, relative strength index (RSI), and Bollinger Bands (BBs). By relying on purely data-driven technical indicators, we avoid the potential pitfalls of using noisy external information.

Our contributions are as follows:

1. We design and implement a Linear Transformer-CNN model that uses a large language model (LLM) to enhance the dataset, effectively integrating both market data and images generated from the market data.

2. We apply a large language model (LLM) for financial technical analysis, using it to enrich the dataset by generating more refined and accurate technical indicators.

3. We evaluate the model on the S&P 500 stock dataset from 2022 to 2023, demonstrating improved accuracy compared to existing models.

This work is motivated by the need for enhanced predictive accuracy in financial markets, particularly when leveraging large-scale language models to generate meaningful insights beyond simple historical data.

4.2 Related Work

In this section, we review the existing literature related to stock price prediction, focusing on traditional machine learning models, deep learning approaches, and recent advancements in LLM-based methods for financial prediction.

4.2.1 Traditional Machine Learning Models for Stock Price Prediction

Traditional machine learning models, such as Linear Regression, Support Vector Machines (SVMs), and Random Forests (RFs), have long been used for stock market prediction. These models rely heavily on time-series data and technical indicators. However, these techniques often struggle with capturing non-linear patterns inherent in financial data. Moreover, traditional models typically require extensive feature engineering, which is both labor-intensive and prone to errors (Riva et al., 2018).

In (Ince & Trafalis, 2008), the authors used the Support Vector Machine (SVM) model to predict the short-term trend of stock prices and compared it with the experimental results of multilayer perceptron (MLP) and Autoregressive Integrated Moving Average (ARIMA) models. They found that the experimental effect of the Support Vector Machine was better. Wang, Li, and Bao (S. Wang et al., 2017) improved the traditional SVM model by introducing the Novel Advanced Fuzzy Support Vector Machine (NA-FSVM) model to predict the trend of NASDAQ and S&P stock prices, with experimental results verifying the method's effectiveness. Similarly, Panwar et al. (Panwar, Dhuriya, Johri, Yadav & Gaur, 2021) utilized Linear Regression and Support Vector Machines to predict stock prices, concluding that Linear Regression outperformed SVM in this context.

4.2.2 Deep Learning Methods for Stock Price Prediction

With the advent of deep learning, models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) have revolutionized stock price prediction by capturing complex patterns in historical data.

The introduction of LSTM by Hochreiter and Schmidhuber has proven particularly effective in time-series prediction, addressing the issue of vanishing gradients commonly faced by traditional RNNs (Hochreiter & Schmidhuber, 1997). LSTM models are capable of learning long-term dependencies, which makes them well suited for financial time-series forecasting.

On the other hand, CNNs have been applied to extract features from visual financial representations like candlestick charts. Yadav et al. demonstrated that combining CNNs with LSTMs (CNN-LSTM models) significantly improved prediction accuracy by leveraging both spatial and temporal features of stock market data (Yadav, Yadav & Saini, 2022). Additionally, Transformer models, introduced by Vaswani et al., have shown remarkable performance in capturing long-term dependencies in sequential data, offering another powerful approach for stock price prediction (Vaswani et al., 2017).

Recent advancements have seen the development of hybrid models that integrate multiple deep learning architectures to enhance predictive performance. Bao, Yue, and Rao (Bao et al., 2017) combined wavelet transforms, stacked autoencoders, and LSTM models (WSAEs-LSTM) for stock price forecasting, demonstrating superior results compared to single models like RNNs and LSTM models. An attention-enhanced LSTM model has been proposed to extract significant temporal features for stock trend prediction, enabling the model to focus on the most informative parts of financial time series. Kulshreshtha et al. (Kulshreshtha et al., 2020) introduced a hybrid ARIMA-LSTM framework that combines statistical and deep learning models to capture both linear and nonlinear characteristics of stock market time series.

Additionally, Liu et al. (2019) presented a corporate knowledge graph embedding system that incorporates related corporate news and combines stock news sentiment with market data features using a Gated Recurrent Unit (GRU) model for stock price prediction, further showcasing the versatility of deep learning in financial forecasting

(J. Liu et al., 2019).

Moreover, studies like Jiang's have reviewed applications of deep learning models, showing that RNNs, LSTMs, and CNNs are more capable than traditional models when predicting complex financial data, especially in dynamic market environments (Jiang, 2021).

4.2.3 Stock Price Prediction with Large Language Models (LLMs)

Recently, large language models (LLMs), such as GPT-3 and FinGPT, have emerged as a promising tool in stock market prediction, particularly for analyzing textual data like news articles, analyst reports, and social media sentiment. These models leverage vast amounts of unstructured data to generate technical indicators and analyze market sentiment, which can be challenging for traditional models.

Brown et al.'s work on GPT-3 highlights the capabilities of LLMs in few-shot learning scenarios, where they can analyze textual data effectively with minimal examples (Brown et al., 2020). Meanwhile, specialized models like FinBERT, which are fine-tuned for financial sentiment analysis, have proven useful for incorporating non-numerical data into stock prediction models. For example, Li et al. utilized FinBERT alongside LSTM networks to predict stock price movements based on news sentiment, demonstrating improved accuracy in predicting short-term price changes (Halder, 2022).

Moreover, Deng et al. (Deng, He, Hu & Yiu, 2024) explored enhancing few-shot stock trend prediction using LLMs, showing that incorporating LLM-based sentiment analysis into stock prediction models leads to significantly better performance, especially in uncertain market conditions.

The combination of LLMs with traditional stock price prediction methods holds great potential in leveraging textual data, which traditional and even deep learning models often fail to incorporate effectively (Darapaneni et al., 2022; Zhu et al., 2024).

4.3 Materials and Methods

4.3.1 Data Preparation

The dataset used in this study includes stock price and volume data for the S&P 500 index from 2022 to 2023, obtained via Tushare. This dataset comprises daily closing prices and trading volumes for each stock, which are converted into logarithmic returns to stabilize variance and enhance prediction quality.

4.3.2 Data Preprocessing

To improve the predictive accuracy and robustness of the proposed model, a comprehensive data preprocessing pipeline was employed. The pipeline includes standardization, normalization, handling of missing values, and time-series stabilization.

Standardization and Normalization

Standardization and normalization are essential preprocessing steps to ensure that input features are on the same scale, preventing discrepancies caused by different magnitudes.

Standardization

Standardization is applied to adjust the data distribution to have a mean of 0 and a standard deviation of 1. This transformation ensures that all features contribute equally to the model training, preventing features with larger magnitudes from dominating the learning process. The formula for standardization is as follows:

$$z = \frac{x - \mu}{\sigma}, \tag{4.1}$$

where x represents the original data, μ is the mean, and σ is the standard deviation of the data.

Normalization

Normalization is used to rescale the data to a fixed range, typically $[0, 1]$. This step is particularly important for ensuring that gradient-based optimization methods converge faster and perform more effectively. The formula for normalization is

$$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}, \tag{4.2}$$

where x_{min} and x_{max} are the minimum and maximum values of the data, respectively.

By applying standardization and normalization, the data become more suitable for training machine learning models, reducing the risk of numerical instability and ensuring faster convergence during optimization. These preprocessing steps also improve the interpretability of the input data by placing features on comparable scales.

These techniques were applied to the logarithmic returns of stock prices and trading volumes to enhance numerical stability and ensure consistent scaling of features.

Handling Missing Values

Calculations involving logarithmic returns and target values can introduce missing values due to lagging or leading operations in the time-series. To address this, all missing values were removed, ensuring a complete and consistent dataset for model training.

Time-Series Stabilization

Financial time-series data are often non-stationary, exhibiting trends and changing variance. To stabilize variance and remove trends, we calculated logarithmic returns for stock prices and trading volumes as follows:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right), \quad v_t = \log\left(\frac{V_t}{V_{t-1}}\right), \tag{4.3}$$

where P_t and V_t denote the closing price and trading volume at time t, respectively.

The target value for prediction was calculated as

$$y_t = \log\left(\frac{P_{t+\Delta t}}{P_t}\right), \tag{4.4}$$

where Δt is the prediction horizon.

4.3.3 Stock Image Data Representation

To capture more nuanced market behavior, we include technical indicators such as moving averages (MAs), relative strength index (RSI), and Bollinger Bands (BBs) as features. These indicators are widely recognized in financial analysis for their ability to reflect market trends, momentum, and volatility, which are critical for making informed trading decisions. The formulas for these indicators are as follows:

Moving Average (MA):

$$\mathrm{MA}_n(t) = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i} \tag{4.5}$$

where n is the window size for the moving average. The MA smooths price fluctuations over the specified window, helping to identify the underlying trend by filtering out short-term noise, which is particularly useful in volatile markets.

Relative Strength Index (RSI):

$$\mathrm{RSI}_t = 100 - \frac{100}{1 + RS_t} \tag{4.6}$$

where RS_t is the relative strength calculated as the ratio of the average gain to the average loss over a given period. The RSI is a momentum oscillator that quantifies the speed and magnitude of price movements, offering a quantitative measure for identifying overbought (RSI > 70) or oversold (RSI < 30) market conditions.

Bollinger Bands (BBs):

$$\text{BB}_{\text{upper}} = \text{MA}_n(t) + k \cdot \sigma_n(t), \quad \text{BB}_{\text{lower}} = \text{MA}_n(t) - k \cdot \sigma_n(t) \tag{4.7}$$

where $\sigma_n(t)$ is the standard deviation over the window size n, and k is typically set to 2. Bollinger Bands are especially useful for highlighting periods of high and low volatility and are widely used to identify potential price reversals or breakout patterns.

These indicators provide a robust foundation for analyzing market behavior. The SMA reveals long-term trends, the RSI highlights momentum shifts, and Bollinger Bands delineate periods of heightened volatility and potential price extremes.

Stock chart images are generated by converting the time-series data into visual representations. Following the method outlined by (T. Kim & Kim, 2019), the stock data are plotted as candlestick charts over a fixed window. These images are augmented with the aforementioned technical indicators, serving as input to a CNN-based model for visual feature extraction.

The rationale for including SMA, BB, and RSI in the candlestick charts is twofold. First, these indicators are essential tools for market participants, providing critical signals for identifying trading opportunities. By incorporating them into the images, the charts emulate how human analysts interpret market conditions. Second, these indicators enrich the visual context of the charts, enabling the CNN to learn spatial patterns associated with trends, momentum, and volatility, which may not be fully captured by numerical features alone.

Standardization of Chart Generation: To ensure consistency and interpretability, the candlestick charts are standardized with a 20-period SMA, Bollinger Bands, and RSI, which are overlayed. These elements provide additional layers of information:

- The SMA smooths out price fluctuations, making long-term trends visually apparent and reducing the impact of short-term noise.
- Bollinger Bands encapsulate price volatility, with the upper and lower bands acting as dynamic support and resistance levels, highlighting potential overbought or oversold conditions.
- The RSI, plotted as a separate oscillator, quantifies momentum shifts and provides a clearer perspective on market extremes.

The chart generation process is automated using a Python-based pipeline. Each chart covers a fixed sequence length (e.g., 30 days) and is split into training, validation, and testing sets. Indicators are computed and overlayed programmatically, ensuring accuracy and reproducibility.

Visual Features for Enhanced Prediction: The generated dataset combines the candlestick images with enriched time-series data, providing a multi-modal representation of market behavior. By integrating numerical and visual features, the model can capture complex patterns that are critical for stock price prediction, such as the following:

- Breakouts or reversals often visible at Bollinger Band boundaries, indicating potential price movements.
- Momentum shifts highlighted by RSI crossovers, providing signals for market entry or exit.
- Trend continuity or divergence seen in SMA trajectories, aiding in the confirmation of trend strength or reversals.

Figure 4.1 illustrates the components of the generated charts. Figure 4.1a shows a candlestick chart with Bollinger Bands and a 20-period SMA, emphasizing price movements and volatility. Figure 4.1b visualizes trading volume as a bar chart, highlighting activity intensity. Figure 4.1c depicts the RSI, identifying overbought or oversold market conditions.

This approach ensures that critical market dynamics are captured effectively, allowing the predictive model to learn both numerical trends and spatial patterns from the stock image data.

4.3.4 LLM Prompt Engineering

To effectively integrate generative AI, structured prompts were designed to guide the model in analyzing the enhanced dataset and providing consistent, interpretable outputs. A sample prompt used in this study is as follows:

> *You are a professional financial analyst. Based on the provided dataset containing technical indicators (SMA, EMA, RSI, MACD, Bollinger Bands), analyze each trading day. For each date, output: (1) the date; (2) an analysis of whether the market shows overbought or oversold signals, bullish or bearish momentum, and any notable price volatility based on Bollinger Bands.*

Using the structured prompts, a large language model (LLM), specifically ChatGPT4o, was employed to perform detailed daily financial analysis. The enhanced dataset, which includes technical indicators such as Bollinger Bands (BBs), Relative Strength Index (RSI), and Moving Averages (MAs), was analyzed day by day. Through prompt engineering, ChatGPT4o interprets the enriched dataset and generates daily analyses that include the following:

1. Identification of daily market trends (e.g., bullish or bearish).

2. Insights into volatility and momentum based on technical indicators.

3. Evaluation of key support and resistance levels for the day.

4. Recommendations for potential trading strategies.

The analyses were stored in a new CSV file with the following columns:

- Date: The specific trading day being analyzed.

- Analysis: A textual summary of the market conditions based on technical indicators.

4.3.5 Vectorization of Textual Insights

To integrate the textual insights into predictive models, FinBERT, a pre-trained transformer model fine-tuned for financial analysis, was used to vectorize the LLM-generated reports. The process involved the following:

1. Tokenizing the daily analyses using FinBERT's tokenizer to ensure compatibility with the model architecture.

2. Extracting the CLS token embedding for each report, representing the overall text.

3. Aggregating the embeddings into a NumPy array for efficient storage and integration into the predictive pipeline.

These vectorized embeddings provide a numerical representation of the qualitative insights, enabling seamless integration with other features such as technical indicators and historical prices.

4.3.6 Advantages of LLM and Prompt Engineering

The integration of structured prompts and generative AI offered several advantages:

- Consistency: Ensured uniform analytical outputs for all trading days, improving interpretability by aligning generated results with standard financial analysis methodologies.
- Actionable Insights: Generated concise and expert-level textual explanations of market conditions by leveraging structured prompts to analyze technical indicators (e.g., SMA, RSI, Bollinger Bands), aiding in better decision-making.
- Enhanced Dataset Representation: Integrated LLM-generated textual features with numerical and technical indicators, enriching the dataset for predictive analysis by providing multi-dimensional perspectives on market conditions.
- Reduced Noise and Improved Reproducibility: Compared to traditional methods of collecting information from social media, which often involve noisy sentiment variability, LLM-generated analyses were shown to be more consistent and reproducible across repeated runs, ensuring reliable inputs for downstream predictive models. Research has shown that social media platforms often amplify noise due to inconsistent or emotionally driven posts from traders, whose sentiments can fluctuate widely based on market rumors or short-term events (Bollen et al., 2011).

4.3.7 Enhanced Dataset for Prediction

The final dataset includes technical indicators and textual insights produced by ChatGPT-4o. These textual insights were derived by instructing the model to act as a financial expert, providing context-specific explanations of market analysis. This multimodal representation facilitates a comprehensive exploration of correlations between numerical and textual features, ultimately improving the quality of the dataset and predictive potential.

4.3.8 Stock Prediction Model Framework

We propose a hybrid model that integrates a Linear Transformer-based time-series analysis with a CNN-based image feature extraction framework for stock price prediction. The framework is further enhanced by leveraging ChatGPT4o, a large language model (LLM), to generate technical indicators from market data, including moving averages (MAs), Relative Strength Index (RSI), and Bollinger Bands (BBs). ChatGPT4o is also utilized for technical analysis, producing enhanced stock data. Additionally, FinBERT is employed to transform the output from the technical analysis into vectors, which are incorporated into the prediction process.

The structure of the proposed framework is illustrated in Figure 4.2. In the following sections, we explain each component of the framework as depicted in Figure 4.2 in detail.

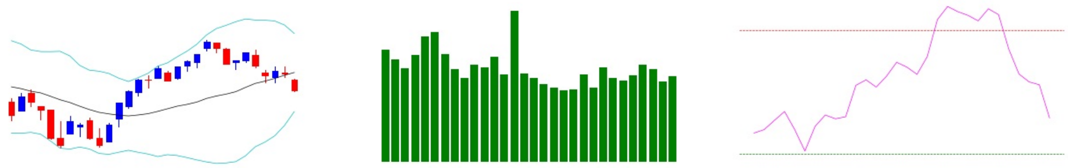
The model consists of three main branches:

- Time-Series Branch: This branch leverages the Linear Transformer model to process historical stock price data. Unlike the standard Transformer, which relies on softmax-based attention, the Linear Transformer approximates the self-attention mechanism using a kernel function to improve computational efficiency while maintaining long-range dependencies in the sequence. Positional encoding is still applied to capture temporal patterns in the stock data.

  The linearized self-attention mechanism is defined as follows:

  $$\text{Attention}(Q, K, V) = \phi(Q)\left(\phi(K)^{\intercal} V\right)$$

  where Q, K, and V represent the query, key, and value matrices, respectively. $\phi(\cdot)$ is the kernel function that ensures non-negative attention values.



(a) (b) (c)

Figure 4.1: Stock data visualizations. (**a**) Candlestick chart (blue candle is bullish and the red candle is bearish) with Bollinger Bands (cyan curve) and 20-SMA (black curve). (**b**) Volume data (green candle) representation. (**c**) RSI (pink curve) indicator chart (red dotted line is overbought and green dotted line is oversold).

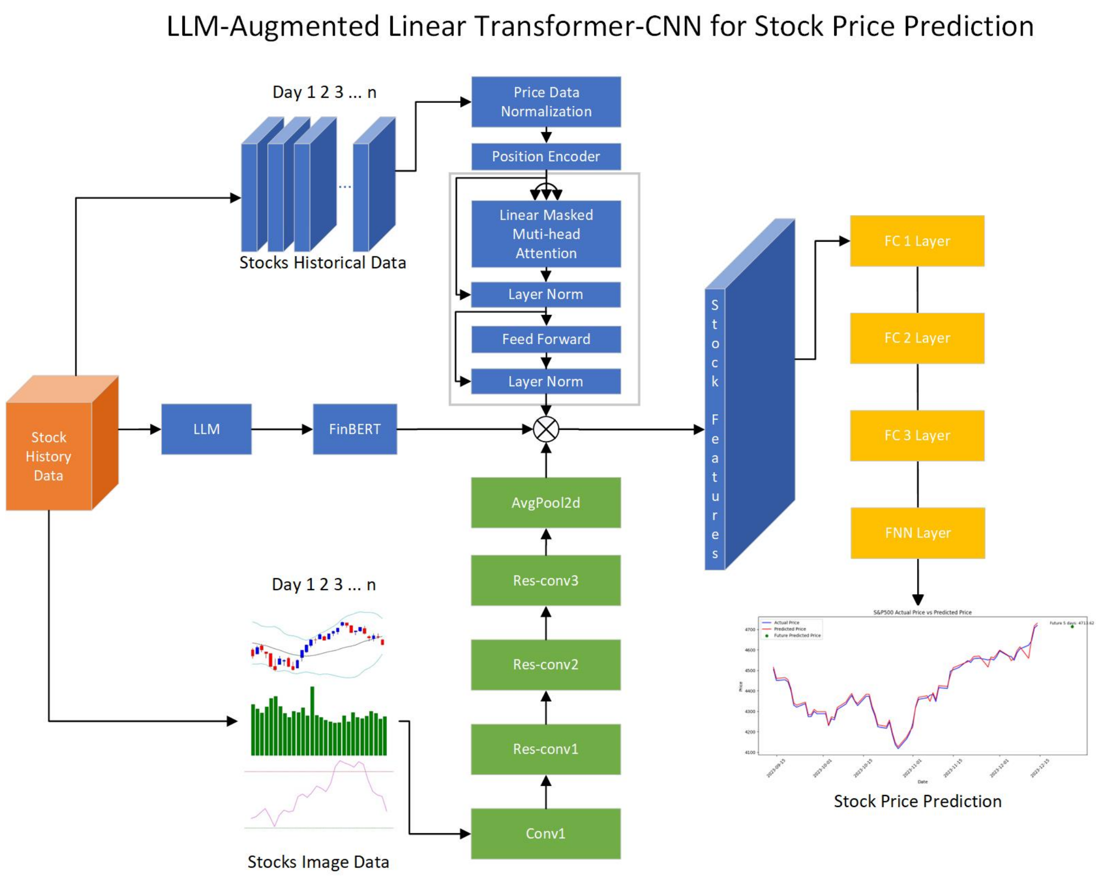


Figure 4.2: LLM-Augmented Linear Transformer–CNN Framework.

The kernel function $\phi(x)$ used in the Linear Transformer is defined as follows:

$$\phi(x) = \text{ReLU}(x) + \epsilon$$

where ϵ is a small constant to avoid division by zero, ensuring stable numerical computations.

Multi-head attention is employed to capture relationships across different representations:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

where each attention head is computed as

$$\text{head}_i = \phi(QW_i^Q)\left(\phi(KW_i^K)^\top VW_i^V\right).$$

Residual connections and layer normalization are applied to the output of each layer to avoid gradient issues:

$$h_1 = h_0 + \text{MultiHead}(h_0, h_0, h_0)$$

$$h_2 = \text{LayerNorm}(h_1) = \frac{h_1 - \mu}{\sqrt{\sigma^2 + \epsilon}}\gamma + \beta$$

After passing through a feed-forward network,

$$h_3 = h_2 + \text{ReLU}(h_2W_1 + b_1)W_2 + b_2.$$

The final representation is obtained after applying another layer normalization:

$$h_4 = \text{LayerNorm}(h_3)$$

- LLM for Daily Analysis and FinBERT for Embedding: To enhance the quality of the input features, we employ ChatGPT4o as a professional financial analyst to produce daily summaries of the stock market. Each summary provides an in-depth analysis based on technical indicators such as MA, RSI, and BB. The LLM-generated analysis focuses on identifying market trends, potential support and resistance levels, and trading opportunities for each day.

  The daily textual analysis is then transformed into numerical vectors using FinBERT, a pre-trained transformer model designed for financial analysis. FinBERT processes the LLM-generated text and extracts contextual embeddings using its [CLS] token representation. These embeddings capture nuanced financial insights and are integrated as additional features into the overall prediction framework.

  The workflow for integrating LLM and FinBERT is as follows:

  - ChatGPT4o analyzes the calculated technical indicators from the stock dataset, acting as a professional financial analyst. Based on this analysis, it generates detailed daily textual summaries.
  - The generated summary is passed through FinBERT to produce vectorized embeddings, which represent the contextual analysis of the stock market for each day.
  - These embeddings are concatenated with the output from the Linear Transformer and SC-CNN branches to create a unified feature representation.

- Image Branch: The SC-CNN (Stock Chart-CNN) branch is responsible for extracting image features from candlestick charts, which include Bollinger Bands and moving averages. This branch is inspired by the work of (T. Kim & Kim, 2019); it utilizes a modified convolutional neural network (CNN) optimized for stock chart images. Inspired by the ResNet architecture, we employ residual learning and a bottleneck structure to address the challenges of overfitting and the vanishing gradient problem, which are common when deepening networks.

The residual learning mechanism uses shortcut connections to bypass certain layers, ensuring smoother gradient flow during backpropagation. This is formalized as

$$F(X) = H(X) + X$$

where $F(X)$ is the output of the residual block, $H(X)$ is the learned function, and X is the input. By setting the residual to zero initially, optimization becomes easier, allowing the network to learn more efficiently as the depth increases.

Furthermore, a bottleneck structure is employed to reduce the time complexity and the number of parameters, while still enhancing the network's ability to extract features. The bottleneck block consists of three convolutional layers: 1×1, 3×3, and 1×1 filters. This configuration significantly increases the number of feature maps and reduces computational costs.

The SC-CNN (Stock Chart-CNN) model uses four convolutional layers, followed by residual blocks (res-conv1, res-conv2, and res-conv3), all optimized for stock chart images. The input stock chart images are resized to 112×112 pixels. The architecture includes the following key components:

– Conv1: Initial convolutional layer for basic feature extraction.

– Residual Blocks (res-conv1, res-conv2, res-conv3): These blocks apply the residual learning technique, allowing deeper layers without suffering from degradation or overfitting issues.

– Fully Connected Layers (fc1, fc2, fc3): Following the convolutional layers, fully connected layers perform the final regression or classification tasks, using features extracted by the convolutional and residual blocks.

We modify the original ResNet-50 architecture by adjusting hyperparameters such as the number of convolutional layers and neurons in fully connected layers, and applying a suitable dropout ratio. The resulting SC-CNN is optimized for stock chart data, maintaining adequate depth while avoiding overfitting. The final extracted feature map from the SC-CNN branch is concatenated with the output from the Transformer branch.

- Fusion and Prediction:

  Once the outputs from all three branches the Linear Transformer branch, the SC-CNN branch, and the FinBERT embedding branch are obtained, we concatenate their final representations into a unified feature vector. Specifically,

  - The final representation h_4 from the Linear Transformer branch, which captures temporal patterns from the stock price data and has dimensions d_1.
  - The feature vector from the SC-CNN branch, which extracts spatial features from the stock chart images and has dimensions d_2.
  - The FinBERT-generated embeddings, which encode contextual and sentiment-based insights from the LLM-generated daily analysis and have dimensions d_3.

  The concatenated feature vector, with a total size of $d_1 + d_2 + d_3$, integrates these diverse modalities, creating a comprehensive representation of the stock market data. This unified representation is then passed through a series of fully connected layers for the final prediction. The architecture of the fully connected layers is as follows:

  - FC1: A dense layer with 500 neurons, followed by a ReLU activation function to model complex interactions between the fused features.

- FC2: A dense layer with 100 neurons and ReLU activation, further refining the feature representation.
- FC3: A dense layer with 25 neurons and ReLU activation, reducing the feature dimension while retaining essential predictive signals.
- Output Layer: A single-neuron output layer for regression, predicting the future stock price.

The number of neurons in each fully connected layer was determined empirically to balance model complexity and computational efficiency. To optimize the model, the Huber Loss function is employed, defined as follows:

$$\text{Loss} = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & \text{if } |y_i - \hat{y}_i| \leq \delta, \\ \delta(|y_i - \hat{y}_i| - \frac{1}{2}\delta), & \text{otherwise,} \end{cases}$$

where y_i represents the actual stock price, $\hat{y}_i$ is the predicted stock price, and δ is chosen based on cross-validation to balance sensitivity and robustness. The Huber Loss is chosen for its ability to handle financial data anomalies effectively.

Dropout layers with a rate of 0.5 are applied after each fully connected layer to prevent overfitting. Early stopping is employed during training, monitoring the validation loss with a patience parameter of 10 epochs. This fusion approach effectively combines temporal, visual, and textual information, enabling the model to capture complex patterns and relationships within the stock market data.

The Huber Loss is preferred over Mean Squared Error (MSE) for stock price prediction because it combines the best properties of both MSE and Mean Absolute Error (MAE). While MSE is sensitive to outliers, making the model overly focused on large errors, Huber Loss offers more robustness to outliers. This is crucial in financial data, where anomalies or sudden market shifts can disproportionately affect the model's performance. By limiting the impact of large errors, Huber Loss helps the model generalize better to unseen data. Additionally, Huber Loss maintains sensitivity to small errors, ensuring accurate predictions for less volatile price movements.

To further improve generalization and avoid overfitting, we apply dropout after each fully connected layer and monitor validation performance using early stopping.

4.4 Experiments

4.4.1 Experimental Settings

We conducted experiments on the Standard and Poor's 500 dataset (S&P 500), using stock price data from 2022 to 2023. Each sample consists of a sequence of stock prices over the preceding 30 days (history length), and the model predicts the stock price movement over the next 5 days (step length). The architecture used in this study combines Transformer and CNN components. Specifically, we employ two Transformer layers to capture temporal dependencies in the data, while four CNN layers are used to extract spatial features.

The model was optimized using a learning rate of 0.001, and training was conducted with a batch size of 64. These settings enabled the model to effectively learn from complex stock price patterns and improve its performance in predicting short-term stock price movements.

4.4.2 Compared Baselines

To evaluate the performance and robustness of our proposed model, we conducted a comparative analysis against several widely used baseline models. These baselines were chosen to represent diverse architectures that are commonly employed in stock market prediction tasks, ranging from traditional recurrent neural networks to more recent

Transformer-based models. By including a variety of architectures, we aim to highlight the advantages and limitations of different approaches in handling multi-modal data and capturing the complex temporal dynamics of stock market behavior.

The baseline models are as follows:

- LSTM-only: An LSTM model trained solely on stock market data. LSTM models are widely used for time-series prediction due to their ability to capture long-term dependencies in sequential data.
- Transformer-only: A Transformer model trained solely on stock market data. Transformers are known for their strong performance in capturing global dependencies in sequential data through attention mechanisms.
- Linear Transformer: A Linear Transformer model trained on stock market data. Linear Transformers offer a computationally efficient alternative to traditional Transformers by approximating the attention mechanism while maintaining competitive performance.
- LSTM-CNN: A combined LSTM and CNN model that utilizes both image data (e.g., candlestick charts) and stock market data for stock prediction. This hybrid approach leverages the temporal modeling capability of LSTMs and the feature extraction strength of CNNs.
- Transformer–CNN: A combined Transformer and CNN model trained on both image data and stock market data. This model combines the global attention capabilities of Transformers with the spatial feature extraction power of CNNs, making it well suited for multi-modal data analysis.

The inclusion of these baselines allows us to comprehensively evaluate our model's ability to integrate and analyze multi-modal data, particularly in comparison to traditional time-series models and hybrid architectures. Each baseline focuses on specific strengths, such as temporal sequence modeling or spatial feature extraction, providing a robust framework for assessing the added value of our proposed approach.

4.4.3 Evaluation Metrics

We employed the following metrics to evaluate model performance, each capturing different aspects of prediction error:

- Root Mean Squared Error (RMSE): RMSE, as the square root of MSE, maintains the same unit as the target variable and is particularly useful for penalizing large deviations.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{4.8}$$

- Mean Absolute Percentage Error (MAPE): MAPE measures the average absolute percentage difference between predicted ($\hat{y}_i$) and actual values (y_i), providing an error measure relative to the actual values. It is useful for understanding model accuracy in percentage terms.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100 \tag{4.9}$$

- Mean Absolute Error (MAE): MAE computes the average magnitude of errors, treating all deviations equally. It is more robust to outliers than MAPE.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{4.10}$$

RMSE, MAPE, and MAE provide a comprehensive evaluation of model accuracy, with RMSE highlighting large errors, MAPE offering an error measure in percentage terms, and MAE providing insight into overall prediction magnitude.

4.4.4 Overall Performance Comparison

In this section, we evaluate the performance of our proposed Linear Transformer–CNN model against several baseline models to assess its effectiveness in stock price prediction. Table 4.1 presents a detailed comparison of our model with various baselines, highlighting their respective strengths and limitations.

Table 4.1: Performance comparison between Linear Transformer–CNN model and baseline methods on historical stock data. Results for RMSE, MAPE, and MAE are shown.

Method	**RMSE**	**MAPE**	**MAE**
LSTM	35.48246	0.00852	35.41428
Transformer	25.74058	0.00473	19.67997
Linear Transformer	19.02579	0.00357	14.80893
LSTM-CNN	53.04961	0.00484	19.93675
Transformer–CNN	30.90289	0.00659	27.49392
Linear Transformer–CNN	13.73599	0.00249	11.0302
% Improvement over feat.	61.3%	70.8%	68.9%

From the results presented in Table 4.1, we derive the following key observations:

- LSTM and Transformer Models: Both models are trained exclusively on historical time-series data. While the Transformer significantly outperforms the LSTM model (with an RMSE of 25.74 compared to 35.48), both models fail to fully capture spatial and contextual features that could further improve predictive accuracy. The Transformer's better performance highlights its strength in modeling temporal dependencies in financial data.

- Linear Transformer Model: The Linear Transformer improves upon the standard Transformer with an RMSE of 19.03, MAPE of 0.00357, and MAE of 14.81. This demonstrates the efficiency of linearized self-attention in capturing long-term dependencies in historical stock data, offering a more computationally efficient alternative for time-series modeling.

- LSTM-CNN Model: By combining LSTM and CNN, this model leverages both time-series and spatial data, resulting in better performance compared to standalone LSTM. However, with an RMSE of 53.05 and MAE of 19.94, it is less effective at capturing the interactions between time-series and spatial patterns, highlighting the need for more advanced architectures.

- Transformer–CNN Model: This model integrates a standard Transformer with a CNN, leveraging both historical time-series and spatial data. Although it demonstrates improved performance over the LSTM-CNN model, achieving an RMSE of 30.90 and MAE of 27.49, it still struggles to match the performance of models incorporating more efficient self-attention mechanisms, such as the Linear Transformer.

- Linear Transformer–CNN (Ours): Our proposed Linear Transformer–CNN model achieves the best performance across all metrics, with an RMSE of 13.74, MAPE of 0.00249, and MAE of 11.03. This superior performance highlights the effectiveness of integrating motif-based subgraph structures with Linear Transformer and CNN architectures. The Linear Transformer efficiently captures long-term dependencies in time-series data, while CNN excels at extracting spatial features from stock chart images. This hybrid approach outperforms all baselines, proving its robustness and adaptability for stock price prediction tasks.

These results emphasize the advantage of our hybrid Linear Transformer–CNN framework, which effectively combines diverse feature representations to achieve state-of-the-art predictive accuracy in stock price forecasting. This study underscores the potential of integrating advanced temporal, spatial, and contextual representations for improved financial forecasting models.

4.5 Discussion

The results presented in this study highlight the potential of combining temporal, spatial, and contextual features for stock price prediction. The proposed hybrid Linear Transformer–CNN framework demonstrates significant improvements over baseline models by effectively integrating diverse feature representations. These findings underscore the importance of a multimodal approach in addressing the inherent complexities of financial markets. However, several important aspects warrant further discussion.

4.5.1 Strengths of the Proposed Framework

Our framework achieves superior predictive performance by leveraging the complementary strengths of the Linear Transformer, a CNN, and LLM-generated features. The Linear Transformer efficiently captures long-term dependencies in historical time-series data while maintaining computational efficiency, making it well suited for processing large-scale financial datasets. Meanwhile, the SC-CNN effectively extracts spatial features from candlestick charts, capturing visual patterns such as trends, volatility, and potential market reversals.

The integration of LLM-generated textual analyses, processed into embeddings using FinBERT, provides context-rich insights that extend beyond traditional numerical features. Unlike sentiment-driven social media data, which can often be noisy and inconsistent, these LLM-generated features offer structured and reproducible interpretations of technical indicators. This multimodal integration challenges the Efficient Market Hypothesis (EMH) by demonstrating that patterns and insights beyond numerical data, such as visual and contextual features, can significantly enhance predictive power. Additionally, the findings align with Behavioral Finance theories, highlighting the role of qualitative and visual cues in influencing market participants' decisions.

This approach not only enhances the predictive accuracy and robustness of the framework but also provides analysts with tools to make more informed decisions. For example, candlestick patterns enable intuitive visualization of market trends, while textual insights deliver expert-level interpretations, bridging the gap between quantitative analysis and qualitative judgment.

4.5.2 Limitations and Challenges

Despite its strong performance, the proposed framework has certain limitations:

1. Dependence on High-Quality Input Data: The framework relies heavily on the quality of technical indicators, candlestick chart images, and LLM-generated analyses. Errors or inconsistencies in data preprocessing or prompt design for the LLM may adversely affect the model's performance.

2. Computational Complexity: Although the Linear Transformer reduces the computational burden compared to standard Transformers, the overall framework, including CNNs and LLM-based analysis, remains resource-intensive. Training such a hybrid model requires significant computational resources, which may limit its applicability in real-time scenarios.

3. Limited External Factors: The model focuses primarily on historical market data, technical indicators, and LLM-generated interpretations, excluding external factors such as news sentiment, macroeconomic data, and geopolitical events, which can have significant impacts on stock prices.

4. Interpretability: While the framework provides accurate predictions, its complexity may hinder interpretability. Understanding how each component contributes to the final prediction requires further exploration, particularly for practical deployment in financial decision-making.

4.5.3 Future Directions

Future work can address these limitations by exploring the following directions:

- Error Analysis in Overpricing and Underpricing Scenarios: Conducting an in-depth error analysis in overpricing and underpricing scenarios could help identify systematic biases in predictions. This would provide valuable insights for developing trading strategies that are tailored to specific market conditions, enhancing the practical utility of the framework.

- Incorporating External Information: Integrating external factors such as news sentiment, macroeconomic indicators, and geopolitical events can further enhance the framework's predictive capability. For instance, combining structured LLM-generated insights with sentiment analysis from news articles could improve the model's contextual understanding.

- Explainability: Implementing explainable AI (XAI) techniques to provide insights into the decision-making process of the model can improve trust and adoption in financial applications. For example, techniques such as SHAP (Shapley Additive Explanations) or attention-based visualizations can help analyze the contributions of individual features.

- Lightweight Models: Developing lightweight variants of the proposed framework can reduce computational overhead, enabling real-time predictions and wider applicability in resource-constrained environments.

- Robustness to Noisy Data: Investigating techniques to enhance the framework's robustness against noisy or incomplete data, such as data augmentation or noise-tolerant algorithms, can improve its performance in real-world scenarios.

4.6 Conclusion and future work

In this chapter, we proposed a novel hybrid framework that combines Linear Transformer, a CNN, and LLM-based analysis for stock price prediction. By integrating temporal, spatial, and contextual features, the framework leverages the strengths of Linear Transformer for efficient time-series analysis, SC-CNN for visual feature extraction, and FinBERT for sentiment-based embeddings derived from LLM-generated daily financial analyses. This multimodal approach significantly outperformed traditional models and hybrid baselines on the S&P 500 dataset, achieving state-of-the-art predictive accuracy.

The study demonstrated that incorporating diverse feature representations enhances the model's ability to capture complex patterns in financial data, providing actionable insights for stock price forecasting. The results underscore the importance of combining technical analysis, historical data, and contextual information to improve predictive performance.

While the proposed framework shows significant promise, several challenges remain, including its computational complexity and limited incorporation of external factors. Future work will focus on integrating external data sources, such as news sentiment and macroeconomic indicators, developing lightweight models for real-time applications, and enhancing model interpretability through explainable AI techniques. These advancements will further improve the robustness and practicality of the framework for financial forecasting applications.

Chapter 5

LLM-Augmented Enhanced Graph Transformer for Stock Movement Prediction

Following the multimodal framework of the previous chapter, this final manuscript revisits graph-based learning by proposing a dynamic graph construction method enhanced by LLM-derived semantic information. Unlike Chapter 3, which uses static motifs, this chapter introduces a graph-based model where nodes represent stocks and edges are statically created based on FinBERT embeddings of LLM-generated textual analyses, while node features evolve daily. A Graph Transformer is then used to capture inter-stock relationships in a unified architecture. This chapter represents the most comprehensive integration of semantic, structural, and temporal information, synthesizing the techniques explored in Chapters 3 and 4 and advancing the thesis's overarching goal of improving financial prediction through advanced deep learning and natural language processing techniques.

5.1 Overview

Predicting stock price movements remains challenging due to the complex interactions and dynamics of financial markets. Recent deep learning advances, particularly integrating numerical data with linguistic analysis via large language models (LLMs), have shown promise. This study proposes an LLM-Augmented Enhanced Graph Transformer that combines LLM-generated financial analyses, FinBERT semantic embeddings, and a Graph Transformer to predict daily stock movements for 260 selected S&P 500 stocks in 2024. We construct a static stock relationship graph based on the cosine similarity of aggregated textual embeddings, capturing long-term semantic dependencies while integrating numerical indicators. Experimental results show our approach outperforms traditional time-series models (e.g., LSTM, Transformer, Informer) and graph-based methods (e.g., GCN, GAT), demonstrating the effectiveness of multimodal fusion and graph-based attention. We also discuss computational constraints and the limitations of static graphs, highlighting future directions such as dynamic graph modeling and optimized text processing.

Predicting price movements in global stock markets remains challenging due to their complex and dynamic nature. Recent years have witnessed the adoption of deep learning techniques for stock trend prediction, incorporating various data sources to enhance accuracy (Q. Li et al., 2020). These sources include technical indicators (Merello et al., 2019), market factors (F. Feng et al., 2018), financial reports (Ballings et al., 2015), news articles (A. W. Li & Bastos, 2020), and social media content (H. Wang et al., 2021), each providing distinct insights into market dynamics.

However, traditional time-series approaches often assume independence and identical distribution (i.i.d.), an assumption rarely valid in financial markets (Awartani & Corradi, 2005). Stocks within the same sector typically exhibit stronger mutual correlations (Y. Chen et al., 2018). To address this limitation, recent studies have employed graph-based methods, explicitly capturing inter-stock relationships by representing financial entities as interconnected nodes within complex networks (D. Cheng et al., 2020; Q. Liu et al., 2018; R. Cheng & Li, 2021).

Additionally, textual data has proven valuable for anticipating market shifts (A. W. Li & Bastos, 2020; H. Wang et al., 2021). Large language models (LLMs) have shown exceptional capabilities in generating nuanced, context-rich analyses by summarizing financial information and capturing subtle linguistic signals.

In this study, we propose an innovative framework integrating numerical market indicators with textual sentiment analyses generated by an LLM within a unified graph-based predictive model. Specifically, our contributions include:

- Considering practical constraints such as computational resources, time efficiency, and hardware limitations, we selected 260 representative stocks from the S&P 500 index. To generate concise daily financial analyses for the period from 1 January, 2024, to 31 December, 2024, we deployed the DeepSeek-R1-Distill-Qwen-14B model on a single machine. This setup ensured efficient processing and consistency in textual analysis generation.

- Employing FinBERT, a financial-domain-specific variant of BERT, to convert these textual analyses into high-quality semantic embeddings. FinBERT provides superior embeddings tailored for financial sentiment, capturing nuanced contextual details better than general-purpose models.

- Constructing a static stock relationship graph where each node represents one of the 260 selected stocks. Edges are determined based on cosine similarity between aggregated FinBERT embeddings derived from historical financial analysis texts. This approach effectively captures long-term semantic relationships among stocks, providing a stable structural foundation for modeling market trends.

- Developing an LLM-Augmented Enhanced Graph Transformer architecture integrating multimodal data numerical time-series and textual embeddings. Leveraging transformer-based attention mechanisms, our model captures complex inter-stock interactions and long-range temporal dependencies effectively.

Our proposed approach overcomes key limitations of traditional Graph Convolutional Networks (GCNs) by explicitly modeling higher-order relationships through attention mechanisms. Experimental evaluations demonstrate significant improvements in predictive accuracy, highlighting the effectiveness of our multimodal fusion strategy.

5.2 Related Work

This section provides a comprehensive review of prior research on stock movement prediction, emphasizing conventional machine learning techniques, advanced deep learning models, and their respective contributions to predictive accuracy (including time-series and graph-based models), and recent advancements in leveraging large language models (LLMs) for financial forecasting, especially in multi-modal contexts related to our LLM+Graph Transformer framework.

5.2.1 Traditional Machine Learning Models for Stock Movement Prediction

Conventional machine learning techniques, including Random Forests, Linear Regression, and Support Vector Machines (SVMs), have been extensively applied in stock market forecasting. These approaches depend heavily on feature engineering and often face limitations in modeling the intricate nonlinear relationships present in financial data. Early studies enhanced classical models with additional inputs like investor sentiment or news-based features to improve accuracy (Cakra & Trisedya, 2015). For example, sentiment scores from news or social media have been integrated into regression and SVM predictors (Cakra & Trisedya, 2015) to better gauge market mood. While such approaches showed some improvements, they faced challenges in scalability and adaptability to rapidly changing market conditions (Devadoss & Ligori, 2013). Overall, the limited capacity of traditional ML models to automatically learn hierarchical representations from raw data motivated the shift toward deep learning approaches for stock movement prediction (Vui, Soon, On, Alfred & Anthony, 2013; Di Persio, Honchar et al., 2016).

5.2.2 Deep Learning Methods for Stock Movement Prediction

Deep learning has significantly advanced stock movement prediction by automatically capturing complex temporal patterns and nonlinear relationships. These approaches can be broadly classified into three categories: time-series models, which process historical stock sequences; transformer-based models, which leverage self-attention to capture long-range dependencies; and graph-based models, which model relational structures in financial markets.

Time-Series Models

Traditional deep learning models process stock prices sequentially to learn historical dependencies:

Long Short-Term Memory (LSTM): LSTMs capture long-term dependencies, mitigating vanishing gradient issues and effectively modeling financial time-series trends (W. Zhang, Chen, Miao & Liu, 2022). Researchers have shown that LSTM-based models can outperform traditional models in predicting stock trends by learning from extensive historical data (Fischer & Krauss, 2018; Selvin, Vinayakumar, Gopalakrishnan, Menon & Soman, 2017). For instance, Fischer and Krauss (Fischer & Krauss,

2018) demonstrated that an LSTM could successfully predict daily stock movements in the S&P 500, exceeding the performance of random forests and logistic regression. LSTMs effectively model the sequential nature of prices, but they do not inherently account for relationships between different stocks.

Convolutional Neural Networks (CNNs): CNNs extract local patterns in stock price sequences, identifying short-term trends through hierarchical feature representations (W. Zhang et al., 2022). By using 1-D convolutions on time-series data, CNN-based models can identify common motifs (e.g., sudden spikes or dips) in stock price sequences and technical indicators (Selvin et al., 2017; Lu, Li, Wang & Qin, 2021). Such convolutional filters capture small-scale trends that can be indicative of immediate future movements. CNNs have also been combined with recurrent models (e.g., CNN-LSTM hybrids) to leverage both local feature extraction and long-term sequence modeling, yielding improved predictive performance (Lu et al., 2021). However, like LSTMs, standard CNNs focus on a single stock's historical data and lack a mechanism to directly incorporate cross-asset influences.

Transformer-Based Models

Transformers leverage self-attention to model long-range dependencies in stock markets. *FinBERT* enhances sentiment analysis in financial texts by leveraging domain-specific financial corpora, outperforming traditional sentiment classification models and general-purpose BERT variants, particularly in distinguishing subtle financial sentiment nuances (A. H. Huang, Wang & Yang, 2023; jun Gu et al., 2024). Meanwhile, *Informer* improves long-sequence forecasting through ProbSparse self-attention, reducing computational complexity and enhancing efficiency in financial time-series prediction (Vaswani et al., 2017; Zhou et al., 2021).

Graph-Based Models

Stock markets exhibit relational dependencies, making Graph Neural Networks (GNNs) an effective tool for modeling asset interactions.

Graph Convolutional Networks (GCNs): GCNs leverage graph structures to enhance stock prediction by incorporating relational dependencies. Chen *et al.* (Y. Chen et al., 2018) showed that industry relationship graphs improve forecasting accuracy. Hybrid models like *GCN-GRU* further enhance performance by integrating temporal dependencies (Long et al., 2020), demonstrating the advantage of graph-based approaches over sequential methods.

Graph Attention Networks (GATs): GATs assign adaptive importance to stock relationships, dynamically capturing market dependencies (Velickovic et al., 2017). Feng *et al.* (S. Feng et al., 2022) proposed a relation-aware GAT that enhances stock prediction by learning influential inter-stock correlations, outperforming static graph models.

Hybrid and Advanced GNN Architectures: Advanced models integrate GNNs with sequence learning to enhance stock prediction. Hou *et al.* (Hou, Wang, Zhong & Wei, 2021) introduced ST-Trader, combining a Variational Autoencoder (VAE) with GCN-LSTM to capture spatial and temporal dependencies, improving market trend forecasting. Other approaches employ global graph pooling and higher-order connectivity for better relational modeling (H. Liu, Simonyan, Vinyals, Fernando & Kavukcuoglu, 2017; S. Wu, Sun, Zhang, Xie & Cui, 2022).

Despite progress, deep learning-based stock prediction faces challenges: High computational cost limits real-time trading applications, dynamic stock relationships remain difficult to define, and model interpretability restricts transparency in decision-making. Future research should prioritize efficient graph learning, explainable AI, and adaptive optimization to enhance deep learning's applicability in financial forecasting.

Comparing time-series and graph-based methods underscores the strengths of LSTM, CNN, Transformer, and GNN models in stock prediction.

5.2.3 Large Language Models for Stock Movement Prediction

Large Language Models (LLMs) have become essential instruments in financial forecasting, particularly in stock movement prediction. Unlike traditional models reliant on structured numerical data, LLMs extract insights from unstructured sources such as financial news, earnings reports, and social media sentiment. By leveraging extensive pretraining on diverse datasets, these models develop enriched representations that effectively capture syntactic structures and semantic meaning, enhancing sentiment analysis and predictive accuracy (Guo et al., 2025; J. Chen, Tang, Zhou & Zhu, 2025; Krause, 2025).

FinGPT, a specialized financial LLM, improves sentiment analysis and prediction by leveraging RLSP for real-time feedback and efficient fine-tuning to minimize computational costs (X.-Y. Liu, Wang, Yang & Zha, 2023).

Recent advancements have positioned reinforcement learning as a key technique in improving the reasoning capabilities of LLMs for financial tasks. DeepSeek-R1, introduced in (Guo et al., 2025), employs reinforcement learning (RL) to enhance its ability to perform complex reasoning tasks without requiring extensive supervised fine-tuning, thus improving predictive accuracy in financial tasks. Comparatively, studies indicate that RL-driven LLMs outperform conventional supervised fine-tuned models in decision-making tasks, as they adapt more dynamically to evolving financial contexts (T. Gao, Jin, Ke & Moryoussef, 2025).

One distinguishing advantage of DeepSeek in financial applications is its computational efficiency. While models such as OpenAI's GPT-4 (Achiam et al., 2023) and

Anthropic's Claude (Kumamoto, Yoshida & Fujima, 2023) demonstrate superior accuracy in certain benchmarks, DeepSeek has been recognized for achieving comparable performance at significantly reduced computational costs (Krause, 2025). Unlike conventional AI models with high resource demands, DeepSeek enhances computational efficiency through specialized architectural optimizations, balancing performance and cost-effectiveness (T. Gao et al., 2025).

Despite these advancements, challenges persist in deploying LLM-based financial models. Concerns such as interpretability, computational overhead, and potential biases necessitate further investigation. Additionally, ethical concerns, including financial misinformation and algorithmic bias, underscore the necessity for regulatory oversight as LLM adoption increases within financial decision-making processes. Nevertheless, the promising results of DeepSeek-R1 and similar models suggest that the fusion of LLMs with reinforcement learning presents a viable pathway toward more accurate and adaptive stock market predictions.

5.3 Materials and Methods

This section describes our LLM-Augmented Graph Transformer framework designed for stock trend prediction. We detail the dataset construction, including the collection of historical stock data, generation of technical indicators, and textual analysis via a Large Language Model (LLM). Subsequently, we outline the preprocessing procedures, involving feature extraction, textual embedding via FinBERT (Araci, 2019), and construction of a static adjacency matrix based on textual semantic similarity. Finally, we present the methodology for integrating these multimodal inputs within a Graph Transformer, effectively capturing temporal and inter-stock relationships.

The dataset presented in Table 5.1 and Table 5.2 consists of daily trading data from 260 stocks listed on the S&P 500, covering the period from 2 January 2024, to

Table 5.1: Example of the Enhanced Dataset for Stock Movement Prediction (January 2, 2024 – December 31, 2024)

Date	**Open**	**High**	**Low**	**Close**	**Volume**	**MACD**	**Signal**	**Hist**	**Boll Middle**	**Boll Upper**	**Boll Lower**	**RSI**
2024-01-02	0.112	0.158	0.208	0.013	-0.300	2.506	2.859	-0.292	-0.806	-0.187	-1.145	1.876
2024-01-03	-1.079	-0.250	-0.857	-0.017	0.399	2.343	2.784	-0.500	-0.701	-0.158	-1.002	1.794
2024-01-04	-1.104	-1.276	-0.994	-1.174	0.810	1.881	2.625	-1.188	-0.669	-0.188	-0.909	0.205
2024-01-05	-1.173	-1.203	-1.239	-1.259	-0.353	1.474	2.409	-1.644	-0.651	-0.207	-0.854	-0.803
2024-01-08	-0.730	-0.949	-1.053	-1.237	-0.444	1.142	2.165	-1.876	-0.639	-0.218	-0.818	-1.310

Table 5.2: Corresponding Textual Analysis Generated by LLM for Each Trading Day

Date	**LLM-Generated Analysis**
2024-01-02	Price reversal potential with bearish MACD (Histogram<0); Bollinger Bands tightening.
2024-01-03	Price closed near highs with high volume; Bollinger Bands suggest potential volatility increase.
2024-01-04	Negative MACD signal (Hist<0), bearish Bollinger Bands; RSI suggests oversold conditions.
2024-01-05	Bearish trend; closing below open with moderate volume and declining RSI indicating continued downward momentum.
2024-01-08	Bearish signal; MACD indicates downward momentum, RSI deeply oversold, suggesting possible short-term reversal.

31 December 2024. Each record includes fundamental trading variables such as the opening, highest, lowest, and closing prices, along with trading volume. Additionally, a set of essential technical indicators including the Moving Average Convergence Divergence (MACD), MACD Signal line, MACD Histogram, Bollinger Bands (Middle, Upper, Lower), and Relative Strength Index (RSI) are computed to capture market dynamics and trends.

Beyond numerical indicators, the dataset is augmented with concise textual analyses automatically generated by a Large Language Model (LLM), specifically DeepSeek-R1-Distill-Qwen-14B. These analyses, limited to 20 words per entry, summarize key financial insights derived from the daily indicators, enhancing the dataset with interpretable and contextually relevant information.

The subsequent sections provide a detailed explanation of the computational methodologies used to derive these technical indicators and describe the automated process of generating daily textual analyses using the LLM.

Technical Indicator Computation

Daily technical indicators computed for each stock include Opening price (O), Highest price (H), Lowest price (L), Closing price (C), Trading volume (Vol), Moving Average Convergence Divergence (MACD), MACD Signal line (Sig), MACD Histogram (Hist), Bollinger Bands (Middle band: BM, Upper band: BU, Lower band: BL), and Relative Strength Index (RSI). These indicators are formally defined below:

MACD Indicators:

$$\mathrm{EMA}_{12}(t) = \mathrm{EMA}(C_t, 12), \tag{5.1}$$

$$\mathrm{EMA}_{26}(t) = \mathrm{EMA}(C_t, 26), \tag{5.2}$$

$$\mathrm{MACD}(t) = \mathrm{EMA}_{12}(t) - \mathrm{EMA}_{26}(t), \tag{5.3}$$

$$\mathrm{Sig}(t) = \mathrm{EMA}(\mathrm{MACD}(t), 9), \tag{5.4}$$

$$\mathrm{Hist}(t) = \mathrm{MACD}(t) - \mathrm{Sig}(t), \tag{5.5}$$

where EMA represents the exponential moving average over a given period, and C_t denotes the closing price on day t.

Bollinger Bands:

$$\mathrm{BM}(t) = \mathrm{SMA}(C_t, 20), \tag{5.6}$$

$$\mathrm{BU}(t) = \mathrm{BM}(t) + 2\sigma_{20}(C_t), \tag{5.7}$$

$$\mathrm{BL}(t) = \mathrm{BM}(t) - 2\sigma_{20}(C_t), \tag{5.8}$$

where SMA denotes the simple moving average and $\sigma_{20}(C_t)$ is the 20-day standard deviation of closing prices.

Relative Strength Index (RSI):

$$\Delta C_t = C_t - C_{t-1}, \tag{5.9}$$

$$\text{Gain}_t = \max(\Delta C_t, 0), \tag{5.10}$$

$$\text{Loss}_t = \max(-\Delta C_t, 0), \tag{5.11}$$

$$\text{AvgGain}_t = \frac{1}{14} \sum_{i=t-13}^{t} \text{Gain}_i, \tag{5.12}$$

$$\text{AvgLoss}_t = \frac{1}{14} \sum_{i=t-13}^{t} \text{Loss}_i, \tag{5.13}$$

$$\text{RS}_t = \frac{\text{AvgGain}_t}{\text{AvgLoss}_t}, \tag{5.14}$$

$$\text{RSI}_t = 100 - \frac{100}{1 + \text{RS}_t}. \tag{5.15}$$

LLM-Augmented Technical Analysis

To augment numerical data, concise daily textual analyses were generated using the DeepSeek-R1-Distill-Qwen-14B LLM deployed locally via Ollama. A structured prompt guided each analysis:

"You are a CFA expert. Provide a very concise technical analysis for [date] using the following indicators. Limit your answer to 20 words and use only essential financial terms. Indicators: open, high, low, close, volume, MACD, MACD_signal, MACD_hist, boll_middle, boll_upper, boll_lower, RSI."

Each day's stock indicators were sequentially integrated into this prompt and processed via automated HTTP requests to the LLM API. Specifically, the API request payload was structured as follows:

$$\text{Payload} = \begin{cases} \text{“model": “deepseek-r1:14b",} \\ \text{“messages": [\{“role": “user", “content": } \textit{Prompt}\text{\}]} \end{cases} \tag{5.16}$$

The LLM-generated outputs underwent post-processing to ensure succinctness and consistency. This included the removal of extraneous markup (e.g., <think> tags), replacement of newline characters with spaces, removal of redundant whitespace, and strict truncation to 20 words:

$$\text{Analysis} = \text{truncate}(\text{remove\_tags}(\text{LLM\_response}), 20\,\text{words}) \tag{5.17}$$

A controlled delay of 0.3 seconds between API calls mitigated potential server overload. The finalized textual analyses were appended to each stock's dataset as a dedicated column, saved in CSV format with UTF-8-SIG encoding to ensure broad software compatibility.

Example analyses generated include:

- *"Bearish momentum with negative MACD histogram and declining RSI suggests further downside risk; Bollinger Bands tighten, suggesting range-bound trading possible."*
- *"Negative MACD, oversold RSI (13%), approaching Bollinger Lower Band, suggesting potential near-term reversal after bearish trend."*
- *"Price closed near high, above Bollinger Bands midline (186.59), suggesting short-term upward momentum and volatility."*

These concise analytical narratives provided valuable semantic context to complement the numerical feature set, significantly enriching subsequent predictive modeling tasks.

Normalization of Technical Indicators

Normalization of features is essential for ensuring stable and efficient model training, particularly when dealing with financial time-series data, which often exhibit varying scales and the presence of extreme values. Among the most commonly used normalization techniques, Min-Max and Z-score normalization are widely employed.

Min-Max normalization rescales data to a fixed range, typically $[0, 1]$, and is mathematically defined as:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \tag{5.18}$$

where X_{min} and X_{max} denote the minimum and maximum values of the feature, respectively. Although effective for maintaining relative relationships among data points, Min-Max normalization is highly sensitive to outliers, which are prevalent in financial markets due to sudden price surges, earnings surprises, or external macroeconomic events. These extreme values can significantly distort the feature distribution, leading to compressed data within a limited range.

To mitigate these issues, we employed Z-score normalization, which standardizes each feature by centering it around zero with unit variance:

$$X_{\text{norm}} = \frac{X - \mu_X}{\sigma_X}, \tag{5.19}$$

where μ_X represents the mean and σ_X denotes the standard deviation of the feature. Unlike Min-Max normalization, Z-score normalization is more robust to extreme fluctuations, preserves the underlying distribution of the data, and ensures that all features are on a comparable scale. This is particularly important for stock market indicators, where assets may have significantly different price ranges and volatility levels.

Therefore, Z-score normalization was systematically applied to all numerical indicators in our dataset, as shown in Table 5.1. This ensures a robust and consistent feature scaling process across different stocks and trading periods.

The final dataset, comprising these normalized technical indicators and corresponding LLM-generated textual analyses, was stored systematically to facilitate its seamless integration into the subsequent graph-based prediction modeling framework.

5.3.1 Statement and Definitions

Preliminaries

We formulate stock market movement prediction as a node-level binary classification task on a **static stock relationship graph**. Specifically, we construct a financial graph $G = (V, E)$ comprising 260 stocks (nodes), where edges are determined based on the cosine similarity of aggregated FinBERT embeddings from historical financial analysis texts. Each trading day's node features and labels are independently generated and stored as compressed `.npz` files, ensuring no future information leakage.

Formally, nodes and edges in the graph are defined as follows:

Definition 1 (Node): Each node $v_i \in V$ represents a stock, characterized by a feature vector $x_i^t \in \mathbb{R}^d$ on trading day t. The feature vector comprises daily technical indicators, including Open (O), High (H), Low (L), Close (C), Volume (Vol), Moving Average Convergence Divergence (MACD), MACD Signal Line (Sig), MACD Histogram (Hist), Bollinger Bands indicators (Middle, Upper, Lower), and Relative Strength Index (RSI). Each node is also associated with a binary label y_i^t, indicating the stock's next-day price movement:

$$y_i^t = \begin{cases} 1, & \text{if Close}_{t+1} > \text{Close}_t \\ 0, & \text{otherwise} \end{cases}$$

Definition 2 (Edge): The edge structure of the stock graph is fixed and determined before daily data generation. Edges $e_{ij} \in E$ are static, established if cosine similarity of aggregated textual embeddings surpasses threshold τ. Specifically, each stock has financial analysis texts aggregated over a fixed historical period, which are transformed into semantic vectors using a Large Language Model (LLM). These textual analyses are then embedded into a semantic vector space using a pretrained financial-domain language model (FinBERT). The similarity threshold τ determines the existence of edges, formally expressed as:

$$e_{ij} = \begin{cases} 1, & \text{if } \frac{h_i \cdot h_j}{\|h_i\| \|h_j\|} \geq \tau, \quad i \neq j \\ 0, & \text{otherwise} \end{cases}$$

where h_i and h_j represent the FinBERT embeddings of the aggregated financial analysis texts for stocks i and j. Notably, the adjacency matrix remains static once constructed and does not change daily.

Algorithm for Graph Generation and Data Preparation

To systematically generate the adjacency matrix $A \in \mathbb{R}^{N \times N}$ (where $N = 260$), the daily feature tensor $X_t \in \mathbb{R}^{N \times 1 \times d}$, and node labels $Y_t \in \mathbb{R}^N$, we propose the following algorithm:

[1] Technical indicator data and LLM-generated daily textual analyses for 260 stocks over 252 trading days; pretrained FinBERT model; cosine similarity threshold τ. For each stock s_i, concatenate its textual analyses across all trading days into a combined document T_i. Encode each combined document T_i into a fixed-dimensional vector $h_i \in \mathbb{R}^{d_e}$ using the pretrained FinBERT model:

$$h_i = \text{FinBERT}(T_i)$$

Compute the cosine similarity between every pair of stock embeddings:

$$\text{sim}(h_i, h_j) = \frac{h_i \cdot h_j}{\|h_i\| \|h_j\|}$$

Construct the static adjacency matrix $A \in \mathbb{R}^{N \times N}$ as:

$$A(i,j) = \begin{cases} 1, & \text{if } \mathrm{sim}(h_i, h_j) \geq \tau \quad \text{and} \quad i \neq j \\ 0, & \text{otherwise} \end{cases}$$

Save the static adjacency matrix:

```
np.save('adj.npy', A)
```

For each trading day t, collect the numerical technical indicators for each stock s_i into feature vectors $x_i^t \in \mathbb{R}^d$, and determine binary labels $y_i^t \in \{0,1\}$ based on next-day price movements. Stack the daily features into tensors:

$$X_t \in \mathbb{R}^{N \times 1 \times d}, \quad Y_t \in \mathbb{R}^N$$

Store the daily feature and label tensors in compressed files:

$$\texttt{np.savez\_compressed('[date].npz'}, X_t, Y_t)$$

A static adjacency matrix file (`adj.npy`) capturing long-term semantic relationships among stocks, and daily compressed files containing feature tensors X_t and labels Y_t.

This algorithm results in a static graph capturing long-term semantic relationships among stocks based on aggregated textual analyses. Daily node features and labels are stored independently, ensuring no information leakage across trading days and facilitating consistent and reproducible model training and evaluation.

5.3.2 LLM-Augmented Enhanced Graph Transformer for Stock Movement Prediction Framework

We propose an enhanced Graph Transformer framework augmented by a Transformer-based temporal encoder to predict stock price movements effectively. The model combines a temporal Transformer encoder, responsible for capturing time-series dependencies, and a Graph Transformer for modeling stock correlations through graph-based interactions.

Given a temporal graph $G = (V, E)$ with N nodes (stocks), each node i at time t possesses a d-dimensional feature vector $x_{i,t}$. We employ a Transformer encoder to extract temporal features:

$$X_i^{temp} = \text{TransformerEncoder}(X_i^{(t-T+1:t)}) \in \mathbb{R}^d$$

where $X_i^{(t-T+1:t)} \in \mathbb{R}^{T \times d}$ represents the feature sequence of node i across a window of T timesteps, and X_i^{temp} denotes the temporal embedding for node i.

Subsequently, we apply a linear transformation to align temporal embeddings to the hidden dimension of the graph model:

$$X_i^{(0)} = W_{proj} X_i^{temp}, \quad X_i^{(0)} \in \mathbb{R}^{d_{hidden}}$$

Graph Transformer with Multi-head Attention

The Graph Transformer is then utilized to aggregate information from neighboring nodes using a graph-aware multi-head attention mechanism. Specifically, given adjacency matrix $A \in \mathbb{R}^{N \times N}$, and projected query Q, key K, and value V:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V$$

The graph-aware attention scores between node i and node j for head h are computed as:

$$\alpha_{ij}^{(h)} = \frac{\exp\left(\frac{(q_i^{(h)})^\top k_j^{(h)}}{\sqrt{d_h}}\right) \cdot A_{ij}}{\sum_{l=1}^{N} \exp\left(\frac{(q_i^{(h)})^\top k_l^{(h)}}{\sqrt{d_h}}\right) \cdot A_{il}}$$

where $q_i^{(h)}, k_j^{(h)} \in \mathbb{R}^{d_h}$ are the query and key vectors for nodes i and j, $d_h = d_{hidden}/n_{heads}$, and A_{ij} indicates connectivity.

Node embeddings are updated as follows:

$$z_i^{(h)} = \sum_{j=1}^{N} \alpha_{ij}^{(h)} v_j^{(h)}$$

The combined embedding from n_{heads} attention heads becomes:

$$z_i = \text{Concat}(z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(n_{heads})}) W^O$$

Multiple stacked Graph Transformer layers iteratively refine node embeddings, denoted as:

$$X^{(l)} = \text{GraphTransformerLayer}(X^{(l-1)}, A), \quad l = 1, 2, \ldots, L$$

where $X^{(l)} \in \mathbb{R}^{N \times d_{hidden}}$ denotes node representations at layer l.

Temporal Transformer Encoder

The Temporal Transformer Encoder explicitly models temporal dependencies within the feature sequences of each node. It utilizes standard Transformer encoder layers:

$$X_i^{temp} = \text{TransformerEncoder}(X_i^{(t-T+1:t)}) \in \mathbb{R}^d$$

Specifically, given an input sequence $X^{(t-T+1:t)} \in \mathbb{R}^{T \times N \times d}$, the temporal encoder processes each node's feature sequence individually, yielding a temporally-aware embedding $X^{temp} \in \mathbb{R}^{N \times d}$.

The final node embedding used as input for the Graph Transformer is the output from the last timestep of the temporal encoder.

Prediction Layer

For each node, a final linear classification layer predicts binary stock movement direction (rise or fall):

$$y_i = \text{Softmax}(W_p z_i + b_p), \quad y_i \in \mathbb{R}^2$$

The predicted label $\hat{y}_i$ is determined by the highest probability class:

$$\hat{y}_i = \arg\max_{c \in \{0,1\}} y_i^{(c)}$$

Loss Function and Training

The cross-entropy loss supervises the training process:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=0}^{1} y_i^{(c)*} \log(y_i^{(c)})$$

where $y_i^{(c)*}$ represents the true one-hot encoded label for node i, and $y_i^{(c)}$ is the predicted probability.

Overall Framework

Our comprehensive model pipeline is defined as follows:

$$X_i^{temp} = \text{TransformerEncoder}(X_i^{(t-T+1:t)})$$

$$X_i^{(0)} = W_{proj} X_i^{temp}$$

$$X^{(L)} = \text{GraphTransformer}(X^{(0)}, A)$$

$$y_i = \text{Softmax}(W_p X_i^{(L)} + b_p)$$

This combined temporal-graph approach ensures precise modeling of both temporal dynamics and cross-node interactions, resulting in enhanced prediction accuracy for stock movement.

5.4 Experiments

To comprehensively evaluate the effectiveness of the proposed LLM-Augmented Enhanced Graph Transformer for stock movement prediction, extensive experiments were conducted. The experiments aim to demonstrate that our method, leveraging both structural graph information and LLM-augmented textual embeddings, significantly outperforms baseline models on financial market prediction tasks.

5.4.1 Experimental Settings

In our experiments, we used historical stock price data and technical indicators as features, specifically: open, high, low, close, volume, MACD, MACD signal, MACD histogram, Bollinger Bands (middle, upper, lower), and RSI. Each sample consists of a sequence length of 5 trading days.

Extensive experiments evaluate our model's effectiveness, comparing performance against baseline models (GCN, GAT, Transformer, LSTM, RNN, Informer). Technical

Table 5.3: Performance Comparison of Different Models for Stock Movement Prediction

Model	Accuracy	F1-score	AUC	Precision	Recall
Graph-based Models					
Graph Transformer (Ours)	**0.7010**	**0.6994**	**0.7865**	**0.7049**	**0.7010**
GCN	0.5939	0.5875	0.6472	0.5999	0.5939
GAT	0.6215	0.6155	0.6652	0.6293	0.621
Time Series Models (without Graph)					
Transformer	0.5499	0.5357	0.5582	0.5465	0.5499
RNN	0.5396	0.6195	0.5632	0.5336	0.7382
LSTM	0.5317	0.5764	0.5459	0.5496	0.6060
GRU	0.5486	0.5906	0.5659	0.5488	0.6392
Informer	0.5416	0.6188	0.5510	0.5467	0.7128

indicators and FinBERT embeddings with a cosine similarity threshold of 0.9 form the dataset. Models were trained for 100 epochs on an NVIDIA RTX3090 GPU.

5.4.2 Compared Baselines

To validate our proposed Graph Transformer model's performance, we compared it against several state-of-the-art baselines:

- **Graph-based models:**
  - **GCN (Graph Convolutional Network)**: Aggregates information from graph neighbors via normalized adjacency matrices (Kipf & Welling, 2016).
  - **GAT (Graph Attention Network)**: Utilizes attention mechanisms to adaptively aggregate node representations (Velickovic et al., 2017).
- **Time series models (without Graph structure):**
  - **Transformer**: Standard Transformer encoder with causal masking for temporal sequence modeling (Vaswani et al., 2017).

- **LSTM**: Long Short-Term Memory network, a recurrent neural network variant capable of modeling long-term dependencies (Hochreiter & Schmidhuber, 1997).
- **RNN**: Basic recurrent neural network model capturing temporal dependencies (Schuster & Paliwal, 1997).
- **GRU**: Gated Recurrent Unit, another variant of recurrent neural networks designed to address the vanishing gradient problem (Chung et al., 2014).
- **Informer**: Transformer-based architecture optimized for efficient handling of long sequences (Zhou et al., 2021).

5.4.3 Evaluation Metrics

Evaluation metrics include Accuracy, F1-score, AUC, Precision, Recall, and Loss, each rigorously defined to assess predictive performance.

These metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.20}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.21}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.22}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.23}$$

$$\text{AUC} = \int_0^1 \frac{TPR}{FPR} d(FPR) \tag{5.24}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives respectively. TPR and FPR represent true positive rate and false positive rate.

5.4.4 Overall Performance Comparison

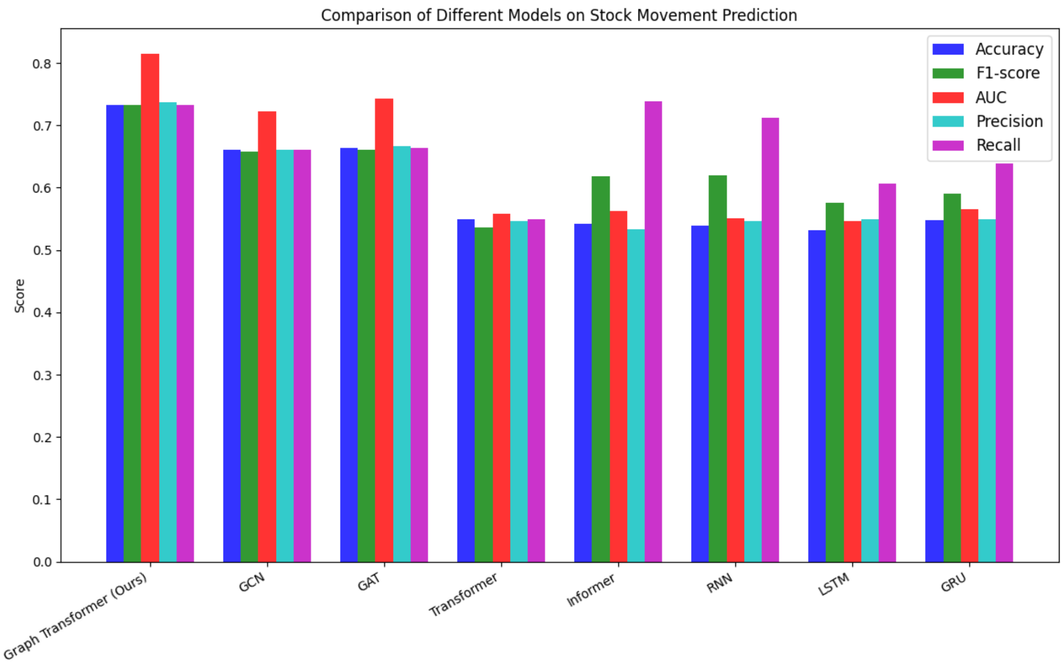


Figure 5.1: Comparison of Different Models on Stock Movement Prediction.

Table 5.3 and Figure 5.1 present a comprehensive comparison of the proposed LLM-Augmented Enhanced Graph Transformer model with other baseline methods. Based on the results, several key observations can be summarized:

- The proposed Graph Transformer demonstrates superior overall performance, achieving the highest accuracy (0.7010), F1-score (0.6994), AUC (0.7865), and precision (0.7049) among all compared models. Although its recall (0.7010) is not the absolute highest, it maintains a strong balance between precision and recall, resulting in the best F1-score. This highlights the effectiveness of combining static graph structures with LLM-enhanced textual embeddings.

- Among graph-based baselines, both GCN and GAT significantly outperform purely temporal models, underscoring the importance of modeling inter-stock relationships beyond individual time series.

- Notably, several temporal models, such as RNN and Informer, achieve relatively high recall values (e.g., RNN recall = 0.7382, Informer recall = 0.7128). This indicates that these models are more aggressive in predicting positive classes (stock price rises), capturing more true positives. However, their lower precision and AUC scores suggest a substantial increase in false positives, meaning they often predict upward movements even when the actual movement is downward. This imbalance reduces overall reliability and hurts performance in real trading applications, where false positives can lead to significant financial losses.

- In contrast, our proposed Graph Transformer model strikes a better precision-recall trade-off. While maintaining a competitive recall, it achieves significantly higher precision and AUC compared to temporal-only models, indicating a more conservative and accurate prediction strategy. This balanced behavior is crucial for financial forecasting tasks, where the cost of incorrect positive predictions (e.g., buying incorrectly) is often higher than missing some positive cases.

- Additionally, the performance gap between the proposed Graph Transformer and time-series models highlights that relying solely on sequential price patterns is insufficient. The integration of graph-based structural priors and semantic financial information substantially enhances the model's ability to understand market behavior.

In summary, although certain RNN or Transformer-based models occasionally achieve higher recall, their overall predictive quality is inferior due to lower precision and AUC. The proposed LLM-Augmented Enhanced Graph Transformer model demonstrates the best comprehensive performance across all key evaluation metrics, validating the advantages of multimodal graph-based learning in stock movement prediction.

5.5 Discussion

Our experimental results demonstrate that integrating LLM-enhanced textual embeddings into a static graph structure significantly improves stock movement prediction performance. By combining numerical technical indicators with rich semantic representations derived from financial text analyses, our model captures both temporal dynamics and inter-stock relationships more effectively than traditional time-series or graph-based methods.

One key advantage of the proposed approach lies in its ability to leverage stable, semantically meaningful relationships across stocks without the need for complex dynamic graph construction. The static graph structure, built on aggregated historical textual similarities, provides a consistent and computationally efficient foundation for graph-based learning while still capturing relevant cross-stock dependencies.

However, several limitations remain. First, the use of a static adjacency matrix may restrict the model's ability to fully reflect rapidly evolving market conditions, particularly during periods of structural shifts or sudden events. Although our framework successfully captures long-term semantic correlations, future extensions could explore periodic dynamic graph modeling techniques, where edge weights or connections are updated at regular intervals (e.g., weekly or monthly) based on accumulated new textual or market data.

Second, the computational cost of generating daily LLM analyses and FinBERT embeddings is non-trivial, especially when scaling to larger stock universes or higher-frequency data. Optimizing text generation pipelines or employing lightweight LLM variants could help mitigate these resource demands.

Finally, while the current model demonstrates strong predictive power, interpretability remains a challenge. Although attention mechanisms offer some insights into feature importance and inter-stock influences, more dedicated explainable AI (XAI) methods would be valuable for understanding the decision-making process, especially in practical financial applications where transparency is critical.

Overall, our work highlights the promise of combining large language models with graph neural networks for financial prediction tasks and sets the stage for future research into more adaptive, efficient, and interpretable frameworks.

5.6 Conclusion

In this chapter, we introduced an LLM-Augmented Enhanced Graph Transformer framework for stock movement prediction, integrating numerical market indicators with LLM-generated textual analyses within a static graph structure based on FinBERT semantic embeddings. By modeling long-term semantic relationships between stocks and capturing temporal market dynamics, our approach significantly outperformed traditional time-series models (e.g., LSTM, Transformer, Informer) and graph-based models (e.g., GCN, GAT) across multiple evaluation metrics.

The combination of multimodal data fusion and graph-aware attention mechanisms enables our model to effectively represent both sequential and relational patterns in financial markets. Experimental results demonstrate that incorporating LLM-enhanced semantic information into graph learning can substantially improve predictive accuracy.

Despite its advantages, our approach faces limitations related to static graph assumptions and computational overhead. Future research directions include exploring dynamic graph modeling to better capture evolving market structures, employing retrieval-augmented generation (RAG) or supervised fine-tuning (SFT) to enhance the quality of textual embeddings, and adopting incremental embedding strategies to improve computational efficiency. Additionally, enhancing model interpretability through explainable AI techniques remains an important avenue for practical deployment.

Overall, our findings reinforce the significant potential of combining large language models with graph neural networks for advancing financial prediction, paving the way for more accurate, adaptive, and transparent stock market forecasting systems.

Chapter 6

Conclusion and Future Work

This chapter concludes the thesis by summarizing the key findings, contributions, and methodological advances across the proposed deep learning frameworks for stock market prediction. It also outlines limitations and discusses potential directions for future research.

6.1 Conclusion

Stock market prediction remains a formidable challenge due to the market's inherent volatility, high dimensionality, and the dynamic interdependence between financial entities. Traditional statistical models and early machine learning techniques often fall short in capturing the complex temporal, structural, and semantic patterns present in financial data. With the rapid advancement of deep learning-particularly in the domains of Transformer models, Graph Neural Networks (GNNs), and Large Language Models (LLMs)-there is growing potential to address these challenges through data-driven, multimodal, and relational learning strategies.

This thesis has explored three core research problems central to advancing the state of stock prediction:

1. How can both temporal dynamics and structural dependencies in the stock market be jointly modeled using motif-based graph learning?
2. How can multimodal information-including historical prices, candlestick charts, and financial text-be effectively integrated to improve prediction robustness while minimizing reliance on noisy external sources?
3. How can semantic relationships among stocks, extracted from LLM-generated textual insights, be leveraged to construct dynamic graphs that enhance stock movement prediction?

In response to these questions, three complementary deep learning frameworks were proposed and evaluated, each focusing on a distinct yet interrelated dimension of financial forecasting: temporal-structural modeling, multimodal feature fusion, and semantic graph construction.

6.2 Research Contributions and Main Finding

Motif-Based Graph and Transformer Integration for Temporal Structural Modeling

To address the first challenge, this thesis proposed the Motif-based Graph Convolutional Network with Transformer (MGSP). By combining Transformer encoders for capturing long-range temporal dependencies with motif-based graph convolution for encoding high-order structural relations, MGSP successfully models both intra-stock dynamics and inter-stock dependencies. Motifs were extracted from a stock–news bipartite graph, capturing semantically meaningful interactions that are often missed in edge-level graphs. Experimental results demonstrate that MGSP outperforms both GCN and

Transformer baselines, confirming the value of motif-enhanced graph structures in temporal financial modeling.

These findings indicate that incorporating motif-aware structures allows the model to capture persistent relational patterns and high-order dependencies often hidden in flat graphs. This structural enrichment contributes not only to improved prediction accuracy but also to a more robust understanding of latent sector-level interactions during regime shifts and high-volatility conditions.

LLM-Augmented Multimodal Deep Learning for Technical Analysis Prediction

In response to the second research question, we developed a multimodal prediction framework that avoids the noise inherent in external narratives by relying exclusively on market-derived data. Using prompt-engineered LLMs (ChatGPT-4o), we generated concise financial commentaries based on technical indicators. These LLM-generated texts were embedded with FinBERT to capture semantic signals, while candlestick and volume chart images were processed using CNNs, and time-series data was modeled with a Linear Transformer. The fusion of semantic, spatial, and temporal information led to a significant improvement in prediction accuracy compared to unimodal baselines, demonstrating the efficacy of integrating LLM-generated technical insights within a multimodal deep learning framework.

The experimental results suggest that LLM-generated textual rationales, when aligned with structured market data, serve as a robust, interpretable, and context-aware signal source that mitigates the noise commonly observed in exogenous financial texts. By aligning language-generated rationales with structured market data, the model bridges the gap between human-like reasoning and quantitative analysis, enhancing both accuracy and model transparency in high-frequency forecasting environments.

Graph Construction via LLM-Enhanced Semantic Embeddings

The third contribution focused on constructing adaptive stock graphs from semantic relationships derived from LLM-generated text. We proposed the LLM-Augmented Enhanced Graph Transformer (LLM-GT), which utilizes daily financial summaries generated by DeepSeek-R1-Distill-Qwen-14B. These summaries, encoded using FinBERT, were used to construct stock relation graphs via cosine similarity, enabling the modeling of latent inter-stock dependencies not captured by traditional industry-based graphs. A Graph Transformer architecture then aggregated both semantic and numerical features using multi-head attention. Our empirical results on the 2024 S&P 500 dataset show that LLM-GT consistently outperforms graph- and sequence-based baselines, illustrating the power of LLMs in dynamic relational modeling for financial forecasting.

We observed that by embedding semantic dynamics into structural representations, the model offers improved adaptability to market sentiment shifts and demonstrates enhanced generalization across varying temporal regimes pointing toward a new class of semantically informed graph learning frameworks for financial AI.

6.3 Future Work

While the proposed frameworks demonstrate strong empirical performance, several limitations and directions for future work remain:

Computational Complexity

The reliance on large scale language models and attention based architectures introduces substantial computational overhead, which may limit deployment in real time or resource constrained environments. Future work could explore model compression techniques, such as knowledge distillation, quantization, or lightweight transformer variants.

Market Generalizability

Our models have been evaluated primarily on the S&P 500 dataset. Their applicability to other markets-such as emerging economies, cryptocurrencies, or non-English stock exchanges-remains an open question. Cross market validation and domain adaptation strategies could further establish model robustness.

Interpretability and Transparency

Despite leveraging attention mechanisms, the overall frameworks still operate as black-box models, which limits interpretability in high-stakes financial contexts. Future studies may incorporate explainable AI techniques (e.g., SHAP, counterfactual explanations, attention visualization) to improve trustworthiness and regulatory compliance.

Graph Construction Strategy

The current graph construction process relies on cosine similarity of FinBERT embeddings, which may oversimplify the nuanced relationships between stocks. Future work could investigate multi relational graphs, hypergraph modeling, or causal discovery techniques to uncover more granular semantic structures.

Incorporation of Macroeconomic and ESG Factors

The present models focus on market derived data such as prices and technical indicators. Integrating macroeconomic variables, earnings reports, environmental social governance (ESG) data, and real time event information could offer additional predictive value and increase resilience under structural market shifts.

6.4 Closing Remarks

This thesis contributes to the financial forecasting literature by introducing three novel deep learning frameworks that incorporate motif-graphs, multimodal representations, and LLM-generated financial insights. Collectively, these approaches address key limitations in modeling temporal, structural, and semantic aspects of stock market data.

The findings validate the utility of motif-graph learning, the interpretive power of LLM-driven technical analysis, and the potential of semantic embeddings for graph construction. By bridging advanced language modeling with financial graph learning, this research lays the groundwork for future intelligent financial systems that are not only accurate but also interpretable, adaptive, and aligned with real world complexities. These contributions not only advance the state of stock prediction but also open promising directions for next generation financial intelligence.

The reflections below represent the author's personal outlook on the broader societal implications of this research.

As artificial intelligence becomes increasingly integrated into economic and societal infrastructures, it is plausible that many traditional forms of labor will undergo profound transformation or eventually become obsolete. In such a future, underpinned by consensus mechanisms and decentralized technologies like blockchain, financial markets may evolve beyond mere capital allocation mechanisms into systems that represent collective intent and societal direction.

Rather than solely pursuing profit, investment could become a participatory act-akin to casting a vote for technological advancement, social development, or industrial growth. Stock transactions, in this vision, may reflect not just economic value but a deeper alignment with shared human priorities.

The true potential of AI lies not in displacing humanity, but in liberating it-from repetitive work, structural inefficiencies, and systemic inequality. In this envisioned future, human roles may shift from task execution to the strategic guidance of collective progress. Consensus, both computational and societal-could emerge as the cornerstone of this new paradigm.

It is the author's sincere hope that the integration of AI and finance may one day contribute to a society in which prosperity is decoupled from labor, and every individual is empowered to live with dignity, autonomy, and purpose. May peace endure, and may this research, in some small way, support the transition toward a more equitable, sustainable, and human-centered future.

References

Abarbanell, J. S. & Bushee, B. J. (1998). Abnormal returns to a fundamental analysis strategy. *Accounting Review*, 19–45.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... others (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agatonovic-Kustrin, S. & Beresford, R. (2000). Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, *22*(5), 717–727.

Ahmed, N. K., Rossi, R. A., Lee, J. B., Willke, T. L., Zhou, R., Kong, X. & Eldardiry, H. (2020). Role-based graph embeddings. *IEEE Transactions on Knowledge and Data Engineering*, *34*(5), 2401–2415.

Ali, A. (2020). Analyzing macroeconomic indicators in pakistan: Insights from unemployment, inflation, and interest rates. *Journal of Business and Economic Options*, *3*(1), 1–12.

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Ariyo, A. A., Adewumi, A. O. & Ayo, C. K. (2014). Stock price prediction using the arima model. *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 106–112.

Asratian, A. S., Denley, T. M. & Häggkvist, R. (1998). *Bipartite graphs and their applications* (Vol. 131). Cambridge university press.

Awartani, B. M. & Corradi, V. (2005). Predicting the volatility of the s&p-500 stock index via garch models: the role of asymmetries. *International Journal of forecasting*, *21*(1), 167–183.

Ballings, M., Van den Poel, D., Hespeels, N. & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert systems with Applications*, *42*(20), 7046–7056.

Bao, W., Yue, J. & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, *12*(7), e0180944.

Benson, A. R., Gleich, D. F. & Leskovec, J. (2016). Higher-order organization of complex networks. *Science*, *353*(6295), 163–166.

Bollen, J., Mao, H. & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, *2*(1), 1–8.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal* *of Econometrics*, *31*(3), 307–327.

Bollinger, J. (1992). Using bollinger bands. *Stocks & Commodities*, *10*(2), 47–51.

Box, G. E. & Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, *65*(332), 1509–1526.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Brock, W., Lakonishok, J. & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance*, *47*(5), 1731–1764.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., . . . others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Cakra, Y. E. & Trisedya, B. D. (2015). Stock price prediction using linear regression based on sentiment analysis. In *2015 international conference on advanced computer science and information systems (icacsis)* (pp. 147–154).

Campbell, J. Y., Lo, A. W., MacKinlay, A. C. & Whitelaw, R. F. (1998). The econometrics of financial markets. *Macroeconomic Dynamics*, *2*(4), 559–562.

Chan, E. P. (2013). *Algorithmic trading: winning strategies and their rationale*. John Wiley & Sons.

Chen, J., Tang, G., Zhou, G. & Zhu, W. (2025). Chatgpt and deepseek: Can they predict the stock market and macroeconomy? *arXiv preprint arXiv:2502.10008*.

Chen, Q. (2021). Stock movement prediction with financial news using contextualized embedding from bert. *arXiv preprint arXiv:2107.08721*.

Chen, W., Hussain, W., Cauteruccio, F. & Zhang, X. (2024). Deep learning for financial time series prediction: A state-of-the-art review of standalone and hybrid models. *Computer Modeling in Engineering & Sciences*, *139*(1), 187–224. doi: 10.32604/cmes.2023.031388

Chen, X., Cai, R., Fang, Y., Wu, M., Li, Z. & Hao, Z. (2023). Motif graph neural network. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(10), 14833–14847.

Chen, Y., Wei, Z. & Huang, X. (2018). Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th acm international conference on information and knowledge management* (pp. 1655–1658).

Cheng, D., Yang, F., Wang, X., Zhang, Y. & Zhang, L. (2020). Knowledge graph-based event embedding framework for financial quantitative investments. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval* (pp. 2221–2230).

Cheng, D., Zhang, Y., Yang, F., Tu, Y., Niu, Z. & Zhang, L. (2019). A dynamic default prediction framework for networked-guarantee loans. In *Proceedings of the 28th acm international conference on information and knowledge management* (pp. 2547–2555).

Cheng, R. & Li, Q. (2021). Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 55–62).

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... others (2020). Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance*, *1*(2), 223.

Cramer, G., Ford, R. & Hall, R. (1976). Estimation of toxic hazard—a decision tree approach. *Food and cosmetics toxicology*, *16*(3), 255–276.

Darapaneni, N., Paduri, A. R., Sharma, H., Manjrekar, M., Hindlekar, N., Bhagat, P., ... Agarwal, Y. (2022). Stock price prediction using sentiment analysis and deep learning for indian markets. *arXiv preprint arXiv:2204.05783*.

Deng, Y., He, X., Hu, J. & Yiu, S.-M. (2024). Enhancing few-shot stock trend prediction with large language models. *arXiv preprint arXiv:2407.09003*.

Devadoss, A. V. & Ligori, T. A. A. (2013). Forecasting of stock prices using multi layer perceptron. *International journal of computing algorithm*, *2*(1), 440–449.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171–4186).

Di Persio, L., Honchar, O. et al. (2016). Artificial neural networks architectures for stock price prediction: Comparisons and applications. *International journal of circuits, systems and signal processing*, *10*, 403–413.

Dwivedi, V. P. & Bresson, X. (2020). A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*.

Fama, E. F. (1965). The behavior of stock-market prices. *The journal of Business*, *38*(1), 34–105.

Feng, F., Chen, H., He, X., Ding, J., Sun, M. & Chua, T.-S. (2018). Enhancing stock movement prediction with adversarial training. *arXiv preprint arXiv:1810.09936*.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y. & Chua, T.-S. (2019). Temporal relational ranking for stock prediction. In (Vol. 37, pp. 1–30). ACM New York, NY, USA.

Feng, R., Jiang, S., Liang, X. & Xia, M. (2025). Stgat: Spatial–temporal graph attention neural network for stock prediction. *Applied Sciences*, *15*(8), 4315.

Feng, S., Xu, C., Zuo, Y., Chen, G., Lin, F. & XiaHou, J. (2022). Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recognition*, *121*, 108119.

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.

Gao, J. (2023). Research on stock price forecast based on arima-garch model. In *Msieid 2022: Proceedings of the 4th management science informatization and economic innovation development conference, msieid 2022, december 9-11, 2022, chongqing, china* (p. 317).

Gao, T., Jin, J., Ke, Z. T. & Moryoussef, G. (2025). A comparison of deepseek and other llms. *The American Statistician*(just-accepted), 1–19.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., ... others (2025). Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Halder, S. (2022). Finbert-lstm: Deep learning based stock price prediction using news sentiment analysis. *arXiv preprint arXiv:2211.07392*.

Hamilton, J. D. (2020). *Time series analysis*. Princeton University Press.

Hamilton, W., Ying, Z. & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, *30*.

Hastie, T. (2009). *The elements of statistical learning: data mining, inference, and prediction.* Springer.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18–28.

Heckmann, M., Domont, X., Joublin, F. & Goerick, C. (2011). A hierarchical framework for spectro-temporal feature extraction. *Speech Communication*, *53*(5), 736–752.

Henrique, B. M., Sobreiro, V. A. & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoseinzade, E. & Haratizadeh, S. (2019). Cnnpred: Cnn-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, *129*, 273–285.

Hou, X., Wang, K., Zhong, C. & Wei, Z. (2021). St-trader: A spatial-temporal deep neural network for modeling stock market movement. *IEEE/CAA Journal of Automatica Sinica*, *8*(5), 1015–1024.

Huang, A. H., Wang, H. & Yang, Y. (2023). Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, *40*(2), 806–841.

Huang, W., Nakamori, Y. & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & operations research*, *32*(10), 2513–2522.

Ince, H. & Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems*, *37*(6), 677–687.

Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, *184*, 115537.

Joachims, T. (1998). *Making large-scale svm learning practical* (Tech. Rep.). Technical report.

jun Gu, W., hao Zhong, Y., zun Li, S., song Wei, C., ting Dong, L., yue Wang, Z. & Yan, C. (2024). Predicting stock prices with finbert-lstm: Integrating news sentiment analysis. In *Proceedings of the 2024 8th international conference on cloud and big data computing* (pp. 67–72).

Katharopoulos, A., Vyas, A., Pappas, N. & Fleuret, F. (2020). Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning* (pp. 5156–5165).

Kelly, B. & Xiu, D. (2023). *Financial machine learning*. Now Publishers.

Khemani, B., Patil, S., Kotecha, K. & Tanwar, S. (2024). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(1), 18.

Kim, R., So, C. H., Jeong, M., Lee, S., Kim, J. & Kang, J. (2019). Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999*.

Kim, T. & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data. *PloS one*, *14*(2), e0212320.

Kipf, T. N. & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krause, D. (2025). Deepseek and fintech: The democratization of ai and its global implications. *Available at SSRN 5116322*.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kulshreshtha, S. et al. (2020). An arima-lstm hybrid model for stock market prediction using live data. *Journal of Engineering Science & Technology Review*, *13*(4).

Kumamoto, T., Yoshida, Y. & Fujima, H. (2023). Evaluating large language models in ransomware negotiation: A comparative analysis of chatgpt and claude.

Lafortune, E. P. & Willems, Y. D. (1993). Bi-directional path tracing.

Lee, M.-C. (2009). Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, *36*(8), 10896–10904.

Lee, S. I. & Yoo, S. J. (2019). Multimodal deep learning for finance: integrating and forecasting international stock markets. *arXiv preprint arXiv:1903.06478*.

Li, A. W. & Bastos, G. S. (2020). Stock market forecasting using deep learning and technical analysis: a systematic review. *IEEE access*, *8*, 185232–185242.

Li, F. et al. (2010). Textual analysis of corporate disclosures: A survey of the literature. *Journal of accounting literature*, *29*(1), 143–165.

Li, J. & Meesad, P. (2016). Combining sentiment analysis with socialization bias in social networks for stock market trend prediction. *International Journal of Computational Intelligence and Applications*, *15*(01), 1650003.

Li, Q., Dong, Y., Wang, D., Zhang, L. & Wu, J. (2021). A real-time inertial-aided cycle slip detection method based on arima-garch model for inaccurate lever arm conditions. *GPS Solutions*, *25*(1), 26.

Li, Q., Tan, J., Wang, J. & Chen, H. (2020). A multimodal event-driven lstm model for stock prediction using online news. *IEEE Transactions on Knowledge and Data*

Engineering, *33*(10), 3323–3337.

Li, W., Bao, R., Harimoto, K., Chen, D., Xu, J. & Su, Q. (2021). Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 4541–4547).

Light, J. O. (2009). *Financial statement analysis* (Vol. 45). Wiley Online Library.

Liu, H., Simonyan, K., Vinyals, O., Fernando, C. & Kavukcuoglu, K. (2017). Hierarchical representations for efficient architecture search. *arXiv preprint arXiv:1711.00436*.

Liu, J., Lu, Z. & Du, W. (2019). Combining enterprise knowledge graph and news sentiment analysis for stock price prediction.

Liu, Q., Cheng, X., Su, S. & Zhu, S. (2018). Hierarchical complementary attention network for predicting stock price movements with news. In *Proceedings of the 27th acm international conference on information and knowledge management* (pp. 1603–1606).

Liu, X.-Y., Wang, G., Yang, H. & Zha, D. (2023). Fingpt: Democratizing internet-scale data for financial large language models. *arXiv preprint arXiv:2307.10485*.

Liu, Y., Zeng, Q., Ordieres Meré, J. & Yang, H. (2019). Anticipating stock market of the renowned companies: A knowledge graph approach. *Complexity*, *2019*(1), 9202457.

Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management, Forthcoming*.

Lo, A. W., Mamaysky, H. & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The journal of finance*, *55*(4), 1705–1765.

Long, J., Chen, Z., He, W., Wu, T. & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in chinese stock exchange market. *Applied Soft Computing*, *91*, 106205.

Lu, W., Li, J., Wang, J. & Qin, L. (2021). A cnn-bilstm-am method for stock price prediction. *Neural Computing and Applications*, *33*, 4741–4753.

Ma, X., Li, X., Feng, W., Fang, L. & Zhang, C. (2023). Dynamic graph construction via motif detection for stock prediction. *Information Processing & Management*, *60*(6), 103480.

MATTHEW, F. D. (2021). *Machine learning in finance: From theory to practice.* Springer.

Medsker, L. R., Jain, L. et al. (2001). Recurrent neural networks. *Design and Applications*, *5*(64-67), 2.

Menike, L., Dunusinghe, P. & Ranasinghe, A. (2015). Macroeconomic and firm specific determinants of stock returns: A comparative analysis of stock markets in sri lanka and in the united kingdom. *Journal of Finance and Accounting*, *3*(4), 86–96.

Merello, S., Ratto, A. P., Oneto, L. & Cambria, E. (2019). Ensemble application of transfer learning and sample weighting for stock market prediction. In *2019 international joint conference on neural networks (ijcnn)* (pp. 1–8).

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D. & Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, *298*(5594), 824–827.

Monti, F., Otness, K. & Bronstein, M. M. (2018). Motifnet: a motif-based graph convolutional network for directed graphs. In *2018 ieee data science workshop (dsw)* (pp. 225–228).

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G. & Grohe, M. (2019). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 4602–4609).

Nelson, D. M., Pereira, A. C. & de Oliveira, R. A. (2017). Stock market's price movement prediction with lstm neural networks. *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 1419–1426.

Nie, Y., Kong, Y., Dong, X., Mulvey, J. M., Poor, H. V., Wen, Q. & Zohren, S. (2024). A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint arXiv:2406.11903*.

Panwar, B., Dhuriya, G., Johri, P., Yadav, S. S. & Gaur, N. (2021). Stock market prediction using linear regression and svm. In *2021 international conference on advance computing and innovative technologies in engineering (icacite)* (pp. 629–631).

Parray, I. R., Khurana, S. S., Kumar, M. & Altalbe, A. A. (2020). Time series data analysis of stock price movement using machine learning techniques. *Soft Computing*, *24*(21), 16509–16517.

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.

Patel, M., Jariwala, K. & Chattopadhyay, C. (2024). A systematic review on graph neural network-based methods for stock market forecasting. *ACM Computing Surveys*, *57*(2), 1–38.

Pennington, J., Socher, R. & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Picasso, A., Merello, S., Ma, Y., Oneto, L. & Cambria, E. (2019). Technical analysis and sentiment embeddings for market trend prediction. *Expert Systems with Applications*, *135*, 60–70.

Qiu, J., Wu, Q., Ding, G., Xu, Y. & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 67.

Rhea, R. (1993). *The dow theory: An explanation of its development and an attempt to define its usefulness as an aid in speculation*. Burlington, VT: Fraser Publishing Company.

Riva, F., Tognollo, A., Gardumi, F. & Colombo, E. (2018). Long-term energy planning and demand forecast in remote areas of developing countries: Classification of case studies and insights from a modelling perspective. *Energy strategy reviews*, *20*, 71–89.

Rossi, R. A., Ahmed, N. K. & Koh, E. (2018). Higher-order network representation learning. In *Companion proceedings of the the web conference 2018* (pp. 3–4).

Rossi, R. A., Zhou, R. & Ahmed, N. K. (2018). Deep inductive network representation learning. In *Companion proceedings of the the web conference 2018* (pp. 953–960).

Saha, S., Gao, J. & Gerlach, R. (2022). A survey of the application of graph-based approaches in stock market analysis and prediction. *International Journal of Data Science and Analytics*, *14*(1), 1–15.

Saviotti, P. P. & Pyka, A. (2008). Micro and macro dynamics: Industry life cycles, inter-sector coordination and aggregate growth. *Journal of Evolutionary Economics*, *18*, 167–182.

Sawhney, R., Agarwal, S., Wadhwa, A. & Shah, R. (2020). Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 8415–8426).

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, *20*(1), 61–80.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117.

Schuster, M. & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, *45*(11), 2673–2681.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K. & Soman, K. (2017). Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 1643–1647).

Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, *90*, 106181.

Sezer, O. B. & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, *70*, 525–538.

Shiller, R. J. (2009). The new financial order: Risk in the 21st century.

Siami-Namini, S., Tavakoli, N. & Namin, A. S. (2019). A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. *arXiv preprint arXiv:1911.09512*.

Singh, R. & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications*, *76*(18), 18569–18584.

Smola, A. J. & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*, 199–222.

Stott, B., Alsac, O. & Monticelli, A. J. (1987). Security analysis and optimization. *Proceedings of the IEEE*, *75*(12), 1623–1644.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, *62*(3), 1139–1168.

Thakkar, A. & Chaudhari, K. (2021). A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Systems with Applications*, *177*, 114800.

Trafalis, T. B. & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Proceedings of the ieee-inns-enns international joint conference on neural networks. ijcnn 2000. neural computing: New challenges and perspectives for the new millennium* (Vol. 6, pp. 348–353).

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M. & Iosifidis, A. (2017). Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 ieee 19th conference on business informatics (cbi)* (Vol. 1, pp. 7–12).

Tsay, R. S. (2005). Analysis of financial time series. *John Eiley and Sons*.

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*.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. et al. (2017). Graph attention networks. *stat*, *1050*(20), 10–48550.

Vijh, M., Chandola, D., Tikkiwal, V. A. & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia computer science*, *167*, 599–606.

Vrandečić, D. & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. In (Vol. 57, pp. 78–85). ACM New York, NY, USA.

Vui, C. S., Soon, G. K., On, C. K., Alfred, R. & Anthony, P. (2013). A review of stock market prediction with artificial neural network (ann). In *2013 ieee international conference on control system, computing and engineering* (pp. 477–482).

Wang, H., Li, S., Wang, T. & Zheng, J. (2021). Hierarchical adaptive temporal-relational modeling for stock trend prediction. In *Ijcai* (pp. 3691–3698).

Wang, N., Yang, H. & Wang, C. D. (2023). Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *arXiv preprint arXiv:2310.04793*.

Wang, S., Li, G. & Bao, Y. (2017). A novel improved fuzzy support vector machine based stock price trend forecast model. In *Proceedings of the 2017 international conference on innovations in economic management and social science (iemss)* (pp. 144–150). Atlantis Press. doi: 10.2991/iemss-17.2017.144

Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B. & Lin, J. C.-W. (2023). A graph-based cnn-lstm stock price prediction algorithm with leading indicators. *Multimedia Systems*, *29*(3), 1751–1770.

Wu, N., Green, B., Ben, X. & O'Banion, S. (2020). Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*.

Wu, S., Sun, F., Zhang, W., Xie, X. & Cui, B. (2022). Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, *55*(5), 1–37.

Wu, Y., Chen, Y., Yin, Z., Ding, W. & King, I. (2023). A survey on graph embedding techniques for biomedical data: Methods and applications. *Information Fusion*, *100*, 101909.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Yu, P. S. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, *32*(1), 4–24.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i. & Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning* (pp. 5453–5462).

Xu, Y. & Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1970–1979).

Yadav, K., Yadav, M. & Saini, S. (2022). Stock values predictions using deep learning based hybrid models. *CAAI Transactions on Intelligence Technology*, *7*(1), 107–116.

Yang, C., Liu, M., Zheng, V. W. & Han, J. (2018). Node, motif and subgraph: Leveraging network functional blocks through structural convolution. In *2018 ieee/acm international conference on advances in social networks analysis and mining (asonam)* (pp. 47–52).

Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C. & Liu, P. (2022). Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, *202*, 117239.

Zhang, W., Chen, Z., Miao, J. & Liu, X. (2022). Research on graph neural network in stock market. *Procedia Computer Science*, *214*, 786–792.

Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., Xu, R. & Wu, B. (2019). At-lstm: An attention-based lstm model for financial time series prediction. In *Iop conference series: Materials science and engineering* (Vol. 569, p. 052037).

Zhang, Y., Yu, J., Ruan, J., Wang, N., Madanian, S. & Wang, G. (2023). Motif-based graph attention network for web service recommendation. In *Proceedings of the 2023 australasian computer science week* (pp. 143–146).

Zhao, Y., Du, H., Liu, Y., Wei, S., Chen, X., Zhuang, F., … Kou, G. (2022). Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. *IEEE transactions on knowledge and data engineering*, *35*(8), 8559–8571.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H. & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 35, pp. 11106–11115).

Zhu, P., Li, Y., Hu, Y., Liu, Q., Cheng, D. & Liang, Y. (2024). Lsr-igru: Stock trend prediction based on long short-term relationships and improved gru. In *Proceedings of the 33rd acm international conference on information and knowledge management* (pp. 5135–5142).

Zong, C., Wan, J., Cascone, L. & Zhou, H. (2025). Stock movement prediction with multimodal stable fusion via gated cross-attention mechanism. *Complex &*

Intelligent Systems, *11*(9), 396.