UNIVERSITY
OF TECHNOLOGY
SYDNEY

Deep Learning in Financial Time Series Forecasting

by Junxian Zhou

Thesis submitted in fulfilment of the requirements for the degree of

Master of Analytics (Research)

under the supervision of Shoujin Wang

University of Technology Sydney
Faculty of Engineering and Information Technology

January 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Junxian Zhou*, declare that this thesis is submitted in fulfilment of the requirements for the award of *Master of Analytics (Research)*, in the *School of Computer Science, Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior
to publication.

DATE: 13th January, 2025

PLACE: Sydney, Australia

Dedication

To everyone,
who has guided me throughout this journey.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Shoujin Wang, for his unwavering guidance, support, and mentorship throughout the two-year Master of Analysis (research) journey. His expertise and insights have been invaluable to me, and I truly appreciate the countless hours he has dedicated to helping me hone my research.

Special thanks go to the research team, The Data Science Lab, especially Distinguished Professor Cao Longbing and Zhangkai Wu, for their collaboration and assistance with deep learning modelling and in-depth research analysis.

I am deeply grateful to the School of Computer Science for providing me with the resources and environment conducive to academic growth. Additionally, I'd like to acknowledge the financial support from my family, which made this research possible.

On a personal note, I owe a huge debt of gratitude to my family, especially my parents, for their endless love, encouragement, and patience. I am forever grateful to my friends who have stood by me, offering emotional support and timely distractions.

List of Publications

Paper Published (Cited By 3):

- **Junxian Zhou**, Shoujin Wang and Yuming Ou, "Fourier Graph Convolution Transformer for Financial Multivariate Time Series Forecasting," 2024 International Joint Conference on Neural Networks (IJCNN), Yokohama, Japan, 2024, pp. 1-8

Paper Submitted/Under Review (Cited By 4):

- Zhangkai Wu, Longbing Cao, Qi Zhang, **Junxian Zhou**, Hui Chen. (2024). Weakly Augmented Variational Autoencoder in Time Series Anomaly Detection, submitted to the IEEE transactions on neural networks and learning systems (TNNLS)

Draft Paper:

- **Junxian Zhou**, Fine-Tuned Large Language Model for Financial Time Series Forecasting with FTS-TextEmbeddings

Abstract

Financial Multivariate Time Series (Fin-MTS) forecasting is a cornerstone of modern financial analytics, underpinning prudent decision-making and risk management in turbulent markets. Compared with other time-series domains, Fin/MTS exhibit five distinctive features: *(i)* pronounced non-linearity, *(ii)* regime-switching volatility, *(iii)* latent periodic structures such as credit or policy cycles, *(iv)* complex intra- and inter-series dependencies across heterogeneous assets, and *(v)* continual interaction with unstructured information (e.g., news and analyst sentiment). These intertwined characteristics complicate modelling and demand methods that capture both quantitative dynamics and qualitative context.

Despite significant advancements in forecasting methods, existing state-of-the-art models often fall short of addressing the intricacies of Fin-MTS. Specifically, these models face several critical challenges: **(C1)** How can models uncover hidden periodic structures? **(C2)** How can intra-series and inter-series dependencies be modelled simultaneously? **(C3)** How can external, unstructured financial information be effectively integrated? **(C4)** How can scalability and interpretability be ensured?

This thesis introduces two innovative models to address these challenges: the Fourier Graph Convolution Transformer (FreTransformer) and the Fine-Tuned Large Language Model for Financial Time Series Forecasting with FTS-Text Embeddings. The FreTransformer utilises frequency-domain transformations to expose hidden periodicities and employs a novel Fourier Graph Convolution Network to capture intra-series and inter-series dependencies in a unified framework effectively. Complementing this, the Fine-Tuned Large Language Model leverages pre-trained large language models to align structured time-series data with unstructured textual information through the FTS-Text Embedder, while the FinLoss optimisation function enhances core financial metrics. Together, these innovative works improve the accuracy of Fin-MTS forecasting by providing scalable, interpretable, and contextually enriched solutions, establishing a robust foundation for future advancements in financial analytics.

Table of Contents

LIST OF FIGURES

FIGURE **Page**

LIST OF TABLES

Part I

Part I

CHAPTER 1

INTRODUCTION

1.1 Background

Along with the Artificial Intelligence (AI) applications developments across the different industries and worldwide, Multivariate Time Series (MTS) provides AI models unlimited possibility for industry and research prediction in financial markets. Quantitative financial models with AI have exhibited superior performance, both experimental and theoretical, by capturing the high dimensional, heterogeneous, non-stable, non. i.i.d dependencies in the markets. Some global-known Hedge Funds, Renaissance Technologies and Two Sigma, utilise advanced quantitative finance AI models, earning more excess return than traditional trading firms. Even conservative professional investor Warren Buffett, who preferred "old-fashioned intelligence", acknowledged the remarkable capabilities of AI. However, in the financial domain, AI technologies imply a certain level of distrust and unease for professional investors. Modelling the dynamics, subjective, competitive markets and market derivatives (e.g., futures, options, forward) with black-box AI models is arduous indeed. Building an AI deep financial model with trustworthiness and explainability is challenging, especially considering the fat-tail effect in which financial data presents non-Gaussian distribution. My thesis is motivated by the issue of how to build a deep learning model that is more effective, trustworthy, and explainable in finance. In the following, there are succinct descriptions of the challenges faced in deep financial modelling, identifications of the gaps in the current AI finance research, and expressions of my contributions to deep financial modelling.

In financial time series forecasting, traditional models contribute to quantifying and predicting the relationships between different markets. The two widely used families of traditional financial Time Series models are ARIMA, AutoRegressive Integrated Moving Average [8] and GARCH, Generalised AutoRegressive Conditional Heteroskedasticity, [6]. They involve the study of data points collected over time to identify patterns, trends, and relationships. They have been applied to real-world applications, price forecasting, volatility modelling, portfolio optimisation, macroeconomic analysis, and algorithmic trading.

However, such linear autoregression methods have significant limitations due to their linear nature. To capture the highly volatile, unstable, non-linear features in the financial dataset, deep learning models, particularly those that leverage convolutional neural networks (CNNs) or recurrent neural networks (RNNs), are advantageous due to their capacity to learn from large volumes of data and their ability to identify complex patterns and relationships. These models can be further enhanced with techniques such as Long Short-Term Memory (LSTM) units to better understand and predict time-series data, capturing critical temporal dynamics in financial markets.

Specifically, Y.LeCun proposed the LeNet-5 at the end of 20^{th} century, which is recognised as the first successful Convolutional Neural Network (CNN) for application [48]. More recently, advanced models with time series, such as RestNet [33], WaveNet [58], and Temporal convolutional networks (TCN) [47] achieved great process in the time series forecasting as well as in the FTS forecasting domain. For instance, CNNpred [36] simply applied the CNN model to the financial dataset.

A Neural Network especially applicable to recognising patterns in sequential data analysis is termed the Recurrent Neural Network (RNN) [61]. With cyclic connection, RNN is more appropriate for financial time series analysis than the feed-forward neural networks theoretically and technically. In 1997, Hochreiter expounded on Long-Short Term Memory, which addressed the challenge of long sequential data training [35]. With the gating mechanism, LSTM can control the flow of information, keeping the gradients in a suitable range, which helps the model train. Basic LSTM uses three gates: input gate, forget gate and output gate. Gating allows the LSTM to have more operations in the data flow and can choose what information is needed or important. Thus, LSTM overwhelmingly overcomes RNN in a single deep model application and mostly beats the hybrid RNN model [31]. It is worth highlighting that LSTM is much more complicated than the RNN, which might not be computationally efficient, which means some Scenarios showed RNN hybrid models have greater performance [75].

Currently, as the most influential deep learning structure, Transformer has had great success not only in neural language processing but also in time series forecasting. For example, FEDformer [85] has been an innovative network using frequency domain knowledge with the low-rank approximation and Discrete Fourier Transform-based attention mechanism. Autoformer [76], an enhanced version of the Transformer, utilises the Fast Fourier Transform (FFT) to disintegrate the components Query (Q), Key (K), and Value (V), thereby capturing long-term period (secular) dependencies within series through amplitude.

Building on the success of Transformer-based architectures, Large Language Models (LLMs) like GPT-4 and LLaMA3 have shown immense potential in adapting to time series tasks by leveraging their powerful sequence modelling capabilities. Unlike traditional models, LLMs possess a pre-trained knowledge base that can be fine-tuned or reprogrammed to understand temporal patterns without requiring architecture-level modifications. For instance, methods like TIME-LLM [40] utilize prompt engineering and text-to-temporal reprogramming to align input data with the natural language processing paradigm, enabling the model to learn complex temporal dependencies. Additionally, frameworks like CALF [52] have introduced cross-modal fine-tuning to address alignment challenges between textual and time-series tokens, further improving prediction accuracy. These developments demonstrate that LLMs, combined with innovative engineering techniques, can transform time series forecasting by bridging the gap between linguistic reasoning and temporal dynamics.

1.2 Research Motivations and Challenges

Deep Financial Time Series Analysis is a novel topic of research that has evolved from naive time series linear regression methods in statistics to leveraging the power of deep neural networks (DNNs) in artificial intelligence (AI) [13]. Any forecasting model more sophisticated than a multilayer perceptron (MLP) can be categorized as a deep forecaster. A key challenge in modern financial time series forecasting is to model the coupling relationships (e.g., dependency, correlation, and causality), redundancy, and relevancy between financial features (e.g., stock tickers, trading volumes) and indicators in the diverse, multisource financial data streams. This complexity underscores the importance of deep financial analysis, particularly following the 2008 Global Financial Crisis (GFC). It is now essential for asset management, portfolio optimization, and risk mitigation in increasingly interconnected financial markets and products.

Financial time series analysis has evolved to predict future trends and identify patterns and anomalies in underlying assets. These patterns can generally be summarized as secular trends, seasonal variations, cyclical fluctuations, and irregular variations [32]. In deep financial forecasting, research challenges expand to study multiple time series simultaneously, involving diverse assets in a single market, each with different features. In 2020, Prof. Cao Longbing emphasized the importance of cross-market analysis, introducing more sophisticated research questions for financial forecasting and advancing the field [13].

Building on these foundations, Large Language Models (LLMs), such as GPT-4 and LLaMA3, have emerged as transformative tools in financial time series analysis. These models leverage pre-trained knowledge and powerful sequence modelling capabilities to process and analyze complex financial datasets with minimal domain-specific adaptations. LLMs can be fine-tuned or reprogrammed to handle temporal patterns effectively, bridging the gap between natural language processing and time series forecasting. For instance, frameworks like TIME-LLM [40] reprogram time series data into textual formats, enabling alignment with language modeling paradigms and allowing the LLM to learn intricate temporal dependencies. However, practitioners must balance these benefits against practical constraints such as the considerable computational cost of Transformer-based LLMs and their susceptibility to performance degradation during abrupt market regime shifts.

Moreover, frameworks such as CALF [52] have introduced cross-modal fine-tuning strategies to address alignment challenges between textual and time-series data representations. This approach significantly improves prediction accuracy by integrating financial reports, news, and real-time market data into forecasting. These advancements in LLM-based techniques open new avenues for financial forecasting, offering a robust combination of linguistic reasoning and temporal data modelling. This holds promise for tackling complex tasks such as anomaly detection, portfolio optimization, and real-time risk management.

1.3 Research Gaps and Questions in Deep Financial Modelling

Deep financial modelling has been a longstanding research interest, particularly in the domain of time series forecasting. Despite extensive studies and experiments leveraging statistical methods and machine learning [13, 19], several critical research gaps persist.

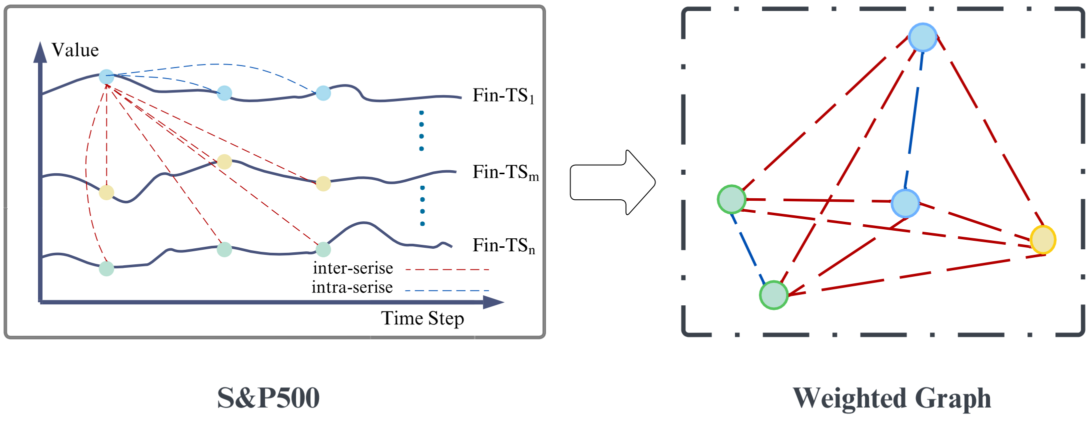
There are many gaps that exist in conventional AI research studies where some existing significant gaps in deep learning will be detailed and discussed in section 4.1. Here are some plain enumerations of the common challenges in deep learning:

1. **Capturing Inter-Series and Intra-Series Dependencies:** Existing deep learning models often fail to adequately model the intricate relationships within and between multiple time series (MTS). In financial data, features within an asset (intra-series) and across different assets (inter-series) are coupled both temporally and spatially. However, conventional methods overlook these correlations, resulting in suboptimal representation learning and predictive performance.
   Research Question 1: How can intra-series and inter-series dependencies be effectively modelled simultaneously?

2. **Adapting to Cyclical and Periodic Features in Financial Markets:** Financial data often exhibit cyclical and periodic behaviors driven by seasonal, economic, or market-specific factors. Conventional models struggle to dynamically capture these recurring patterns, especially in non-stationary and volatile market conditions. This gap limits their robustness in addressing periodic changes in financial time series forecasting.
   Research Question 2: How can models uncover hidden periodic structures in financial data?

3. **Integrating Temporal and Textual Data:** Financial markets are influenced not only by historical price and volume data but also by external drivers such as market sentiment, macroeconomic announcements, and financial reports. Despite advancements, there is limited research on effectively combining temporal data with unstructured textual data to holistically capture these drivers and improve forecasting accuracy.
   Research Question 3: How can external, unstructured financial information be effectively integrated into predictive models?

4. **Leveraging LLMs for Financial Forecasting:** The recent advancements in Large Language Models (LLMs) have opened new possibilities for integrating textual information into financial forecasting. However, there is limited research on how to fine-tune LLMs to align with financial contexts, enabling the extraction of meaningful insights from textual data such as market sentiment, financial reports, and macroeconomic events.



S&P500

Weighted Graph

Figure 1.1: Illustration of the weighted graph with n distinct Fin-TS input. Fin-TS denotes the Financial Time Series. Each feature within every time step of each Fin-TS is represented as a node in a weighted graph.

> **Research Question 4:** How can scalability and interpretability be ensured when applying LLMs to financial forecasting?

1.4 Fourier Graph Convolution Transformer

Financial multivariate time series (Fin-MTS) forecasting presents unique and profound challenges due to the intrinsic nature of financial data. These challenges primarily revolve around the complexities of modelling highly volatile, non-linear, and often non-stationary data, which has both long-term dependencies and abrupt changes. In this thesis, we propose a novel Fourier Graph Convolution Network (FGCN) method to effectively capture the intricate characteristics of Fin-MTS. This approach is specifically designed to model both intra-series and inter-series dependencies in the frequency domain, addressing the non-linearity and non-stationarity of financial data. The details of this architecture and its implementation are illustrated in Fig. 1.1. Based on the characteristics of Fin-MTS and the limitations of current deep learning models, the critical challenges include:

1. **Capturing Hidden Periodicities**: Traditional time-series models often struggle to reveal the hidden periodic characteristics in financial data, such as those driven by economic cycles, credit policies, or market sentiment. This introduces significant

modelling challenges, as ignoring these periodicities can result in information loss and reduced predictive accuracy [10, 12]. Therefore, the challenge lies in developing a robust model that can map financial time series into a frequency domain, where these hidden periodicities can be effectively disclosed and utilized.

2. **Modeling Intra-series and Inter-series Dependencies**: Another critical challenge in Fin-MTS is modelling both the intra-series (within the same series) and inter-series (between different series) dynamic dependencies. Financial markets are highly interconnected, and capturing these dependencies is essential for accurate forecasting. Current models often handle these dependencies separately, leading to limited forecasting performance. The challenge here is to design a model that integrates these dependencies into a unified framework that can reflect the true dynamics of the market [11].

3. **Balancing Model Explainability and Performance**: With deep learning models, especially those involving advanced architectures like transformers or graph neural networks, there is often a trade-off between explainability and predictive power. In financial forecasting, where trust and transparency are critical, it is essential to develop models that not only provide accurate forecasts but also allow financial experts to interpret them. The challenge is in designing a model that balances these two aspects, ensuring that its decision-making process is both understandable and reliable [2, 5].

4. **Handling Non-linear, Non-stationary, and Volatile Data**: Financial data often deviates from the assumptions of traditional time-series forecasting models, including assumptions of linearity, stationarity, and Gaussian distributions. The non-stationary and volatile nature of financial time series requires models that are flexible and adaptive to the evolving market dynamics. This adds a layer of complexity to model design, where the challenge is to create models that can accommodate the diverse characteristics of financial markets without overfitting or losing generalization capabilities [1, 60].

1.5 Fine-Tuned Large Language Model for Financial Time Series Forecasting with FTS-Text Embeddings

Fine-tuned Large Language Models (LLMs) augmented with Financial Time Series-Text (FTS-Text) embeddings represent a cutting-edge approach for addressing the complexities of financial multivariate time series (Fin-MTS) forecasting. These models harness the pre-trained capabilities of LLMs to handle sequential and contextual data while adapting to the specific challenges of Fin-MTS. Figure 1.2 provides a visual representation of the FTS-Text Embedder, a key component in integrating financial time series data with textual information. The diagram illustrates the workflow, where financial time series data, such as the S&P500 dataset, is pre-processed to capture temporal dependencies, while textual data, derived from financial news or reports, is encoded using pre-trained Large Language Models (LLMs). The FTS-Text Embedder subsequently aligns these two distinct data modalities into unified input embeddings, enabling the downstream model to leverage both numerical and contextual insights for enhanced financial forecasting performance. The key contributions and advantages of this approach are as follows:

1. **Bridging Temporal and Textual Domains**: LLMs, pre-trained on extensive textual corpora, excel at processing sequential and contextual information. By integrating Financial Time Series data (FTS) into the textual paradigm, these models align temporal patterns with contextual insights from textual sources such as news, reports, and sentiment analysis. This cross-modal alignment enables the model to incorporate external drivers of market behaviour, providing a more comprehensive understanding of financial time series.

2. **Towards a Unified Forecasting Framework**: By combining FTS and textual embeddings, fine-tuned LLMs create a unified framework that bridges the gap between traditional statistical models and advanced AI techniques. This integration allows the model to analyze diverse information sources simultaneously, offering robust and scalable solutions for tasks such as cross-market analysis, multi-asset forecasting, and risk assessment.

3. **Handling Non-linear, Non-stationary, and Volatile Data**: Financial time series often deviate from traditional assumptions of stationarity, linearity, and Gaussian distributions. Fine-tuned LLMs embed temporal information into tex-

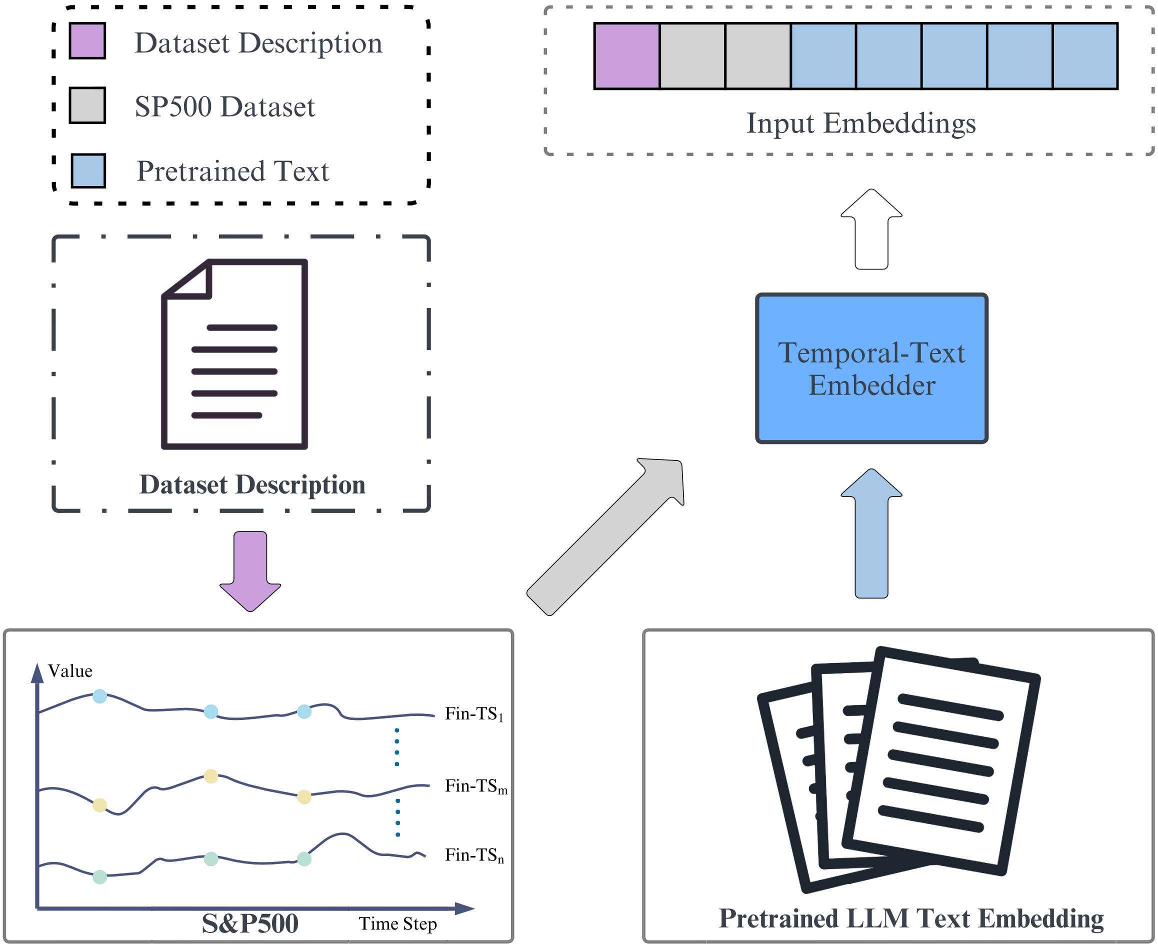


Figure 1.2: Illustration of the FTS-Text Embedder. This high-level diagram demonstrates the process where both the financial time series (Fin-TS) and textual data are combined. The time series data, such as the S&P500 dataset, is processed alongside the pre-trained textual embeddings. These inputs are passed through the FTS-Text Embedder, which aligns the two data types into unified input embeddings..

tual processing paradigms, allowing them to adapt to evolving market conditions dynamically. This flexibility is essential for capturing financial markets' volatile and non-linear characteristics, particularly during periods of economic shocks or regime changes [1, 60].

4. **Explainability and Transparency in Model Predictions**: In financial forecasting, trust and interpretability are crucial. Fine-tuned LLMs offer a transparent framework where predictions can be attributed to specific temporal and textual features. For instance, a forecasted market trend could be explained by linking it to historical price movements and contextual signals extracted from market news or analyst commentary. This balance between explainability and performance ensures that the model's insights are both actionable and trustworthy for financial decision-making [2, 5].

This approach underscores the transformative potential of combining LLMs with FTS-text embeddings in financial forecasting. By addressing the challenges of modelling non-linear dynamics, integrating diverse data modalities, and providing interpretable results, fine-tuned LLMs establish a new benchmark for forecasting methodologies in finance.

1.6 Research Contributions

To effectively address the previously mentioned challenges and gaps, the contributions of this thesis in the deep financial time series forecasting are summarised as follows:

- This thesis proposes a novel transformer design with Graph knowledge embedding to capture the correlation of both inter-series and intra-series to efficiently and interpretably learn the dynamic representation in financial asset time series forecasting.

- This thesis proposes an attention mechanism that combines graph knowledge in the frequency domain, innovatively capturing the financial data representation.

- This thesis integrates Financial Time Series-Text (FTS-Text) embeddings with a fine-tuned Large Language Model (LLM) to leverage both temporal and textual data sources, enabling enhanced understanding of external drivers such as market sentiment, macroeconomic events, and financial reports.

- This thesis introduces a unified framework for multi-asset and cross-market forecasting by aligning dynamic time series patterns with contextual insights extracted from textual information, effectively bridging the gap between time-series forecasting and natural language understanding.

- To handle financial data's non-linear, non-stationary, and volatile characteristics, this thesis develops a robust mechanism that dynamically adapts to market conditions by incorporating external textual drivers and evolving temporal dependencies into the predictive model.

- This thesis emphasises explainability by designing models where the predictions can be transparently linked to specific temporal features and textual signals, ensuring that the results are actionable and interpretable for financial decision-making.

1.7 Thesis Structure

The remainder of this thesis is organised as follows:

Chapter 2 provides the foundational knowledge necessary for understanding the methodologies and models proposed in this thesis. It begins with an overview of financial time series forecasting, emphasizing its unique challenges and characteristics. The chapter then introduces essential mathematical concepts and tools, such as time series analysis, including the decomposition of trends and variations, and statistical modelling. Deep learning fundamentals, particularly neural networks and sequential modelling approaches are also discussed. Furthermore, the principles of graph theory and transformer architectures are presented, laying the groundwork for the novel methods developed in subsequent chapters.

Chapter 3 reviews existing financial time series forecasting research, focusing on three primary areas. First, it examines traditional methods, such as statistical models and early machine learning techniques, identifying their limitations in handling non-linear and non-stationary financial data. Second, the chapter explores modern advancements, including deep learning models and graph-based methods, which aim to capture complex dependencies and patterns in financial data. Third, it reviews the use of transformer architectures in time series forecasting and their integration with financial datasets. The chapter concludes by identifying key gaps in the existing literature, such as limited interpretability, insufficient cross-market analysis, and challenges in leveraging multi-modal data, thus motivating the need for the proposed frameworks.

Chapter 4 presents the Fourier Graph Convolution Transformer (FreTransformer), a novel framework developed to address the challenges of financial multivariate time series forecasting. The FreTransformer integrates graph knowledge embedding to capture inter-series and intra-series dependencies, enabling it to model the dynamic relationships between financial assets. It introduces an innovative attention mechanism tailored for financial data, allowing the model to learn and interpret complex patterns effectively. The chapter provides a detailed theoretical foundation for the model and describes its implementation. Experimental evaluations on real-world datasets demonstrate the superior performance of the FreTransformer compared to existing methods, particularly in capturing volatile and non-linear market dynamics. Ablation studies and sensitivity analyses further validate the robustness of the proposed framework.

Chapter 5 introduces a fine-tuned Large Language Model (LLM) framework specifically designed for financial time series forecasting. The framework incorporates Financial Time Series-Text (FTS-Text) embeddings, enabling the model to process temporal data and textual information, such as financial news and reports, within a unified architecture. The chapter details the key components of the framework, including the FTS-Text Embedder for multi-modal integration, the FinLoss function for optimizing model performance, and the LLM4FT module for improved generalization and interpretability. Extensive experiments are conducted to evaluate the model's performance across various datasets, benchmarks, and financial forecasting tasks. The results highlight the effectiveness of the proposed approach in handling non-linear, non-stationary, and volatile financial data while maintaining a high level of transparency and interpretability.

Chapter 6 summarises the thesis's key contributions to financial time series forecasting. The proposed methods, including the FreTransformer and fine-tuned LLM framework, are highlighted as significant advancements in addressing the unique challenges of financial forecasting. The chapter also discusses the practical implications of these methodologies in real-world financial applications, such as risk management, asset allocation, and portfolio optimization. Finally, potential avenues for future research are outlined, including exploring larger and more diverse datasets, developing scalable architectures for real-time forecasting, and enhancing model interpretability to support decision-making in complex financial environments.

The structure of this thesis, as illustrated in Figure 1.3, provides a clear roadmap of the research and its contributions. The figure outlines the logical progression of the thesis, beginning with foundational knowledge and literature review in Chapters 2 and 3, followed by the development of novel methodologies in Chapters 4 and 5. These chapters focus on the FreTransformer framework and the fine-tuned LLM framework, respectively, addressing key challenges in financial multivariate time series forecasting. Finally, Chapter 6 summarises the contributions, practical implications, and potential future directions, completing the thesis with a comprehensive reflection on its findings. The diagram visually encapsulates the interconnected flow of the thesis, highlighting how each chapter builds upon the previous one to achieve the research objectives.

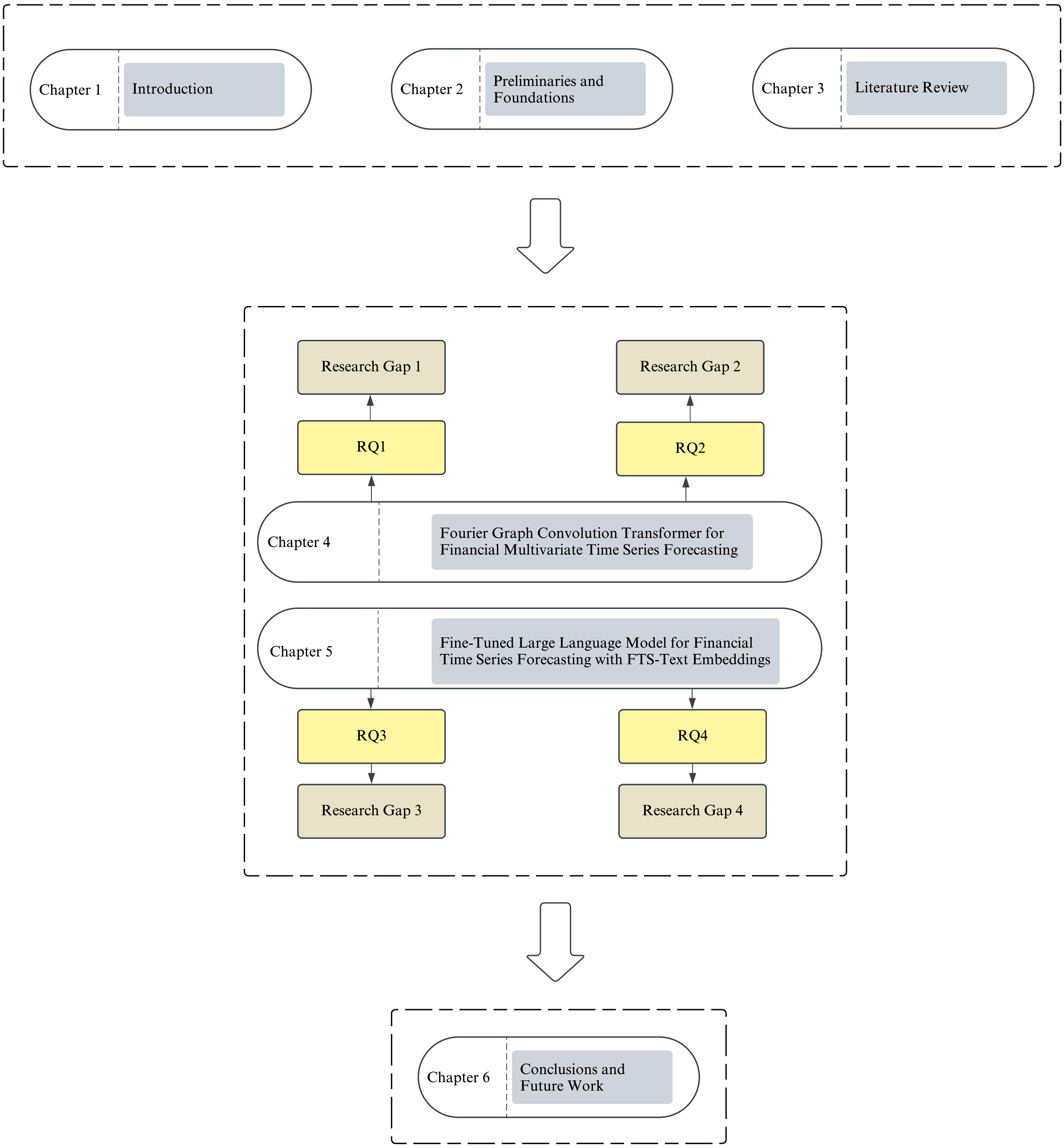

Chapter 1
Introduction
Chapter 2
Preliminaries and Foundations
Chapter 3
Literature Review
Research Gap 1
Research Gap 2
RQ1
RQ2
Chapter 4
Fourier Graph Convolution Transformer for Financial Multivariate Time Series Forecasting
Chapter 5
Fine-Tuned Large Language Model for Financial Time Series Forecasting with FTS-Text Embeddings
RQ3
RQ4
Research Gap 3
Research Gap 4
Chapter 6
Conclusions and Future Work

Figure 1.3: Thesis structure.

CHAPTER 2

PRELIMINARIES AND FOUNDATIONS

This chapter outlines the foundational concepts for this thesis. It begins with an introduction to deep learning models, followed by an overview of key financial time series concepts. Finally, the evaluation metrics relevant to the proposed models in this thesis are discussed.

2.1 Deep Learning Models

In financial time series deep modelling, grasping and selecting the suitable model for different applications is crucial. Whether utilising the deep model directly or applying hybrid models to cope with the problems requires a comprehensive understanding of the popularly used models.

2.1.1 Convolutional Neural Networks

At the end of 20^{th} century, Y.LeCun proposed the LeNet-5, which is recognised as the first successful Convolutional Neural Network (CNN) for application[48]. More recently, advanced models with time series, such as RestNet[33], WaveNet[58], and Temporal convolutional networks (TCN) [47] achieved great process in the financial area.

A traditional CNN has many layers that contain the convolutional layer, pooling layer, fully connected layer, dropout layer and input-output layer with conversion. The CNN convolutional layer equation is given below:

$$V_{xy}^{l} = \sum_{i=-n}^{n} \sum_{j=-m}^{m} K_{i,j} \cdot V_{x+i,y+j}^{l-1}. \tag{2.1}$$

In the Eq(10), V_{xy}^{l} is the output value at (x, y), $V_{x+i,y+j}^{l-1}$ is the input value at $(x+i, y+j)$, $K_{i,j}$ is the convolutional kernel at (i, j), $n \times m$ is the size of the convolutional kernel.

Although CNN was initially designed for image processing, CNN-based models can tackle multiple tasks, such as time series prediction or classification. It is deserving of mention that a possible solution for time series prediction is to transform the time series into an image and then put the image into a CNN based model to analyse[44, 67].

2.1.2 Recurrent Neural Networks

A Neural Network especially applicable to recognising patterns in sequential data analysis is termed the Recurrent Neural Network (RNN)[61]. With cyclic connection, RNN is more appropriate for time series analysis than the feed-forward neural networks theoretically and technically. In 1997, Hochreiter expounded on Long-Short Term Memory, which addressed the challenge of long sequential data training[35]. Moreover, Cho put forward Gated Recurrent Unit (GRU) in 2014 to streamline LSTM module[17]. The simple RNN equations are shown below:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{2.2}$$

$$y_t = g(W_{hy}h_t + b_y), \tag{2.3}$$

where f and g are the activation functions. W is the weight matrix, h_t is the hidden state, y_t is the output, x_t is the input, and b is the bias. In the Eq(11) and Eq(12), we can obtain the continuous matrix transformation will cause the gradient vanishing and exploding problem. During backpropagation, gradients are multiplied many times when the RNN deals with long sequences. Hochreiter and Schmidhuber's seminal work [35] introduced the Long Short-Term Memory (LSTM) model, revolutionizing the handling of long-term dependencies in sequential data. The paper addressed the vanishing gradient problem inherent in traditional Recurrent Neural Networks (RNNs) by introducing a gated cell mechanism. The key components-input gate, forget gate, and output gate-enabled the selective retention and updating of information, allowing the model to learn patterns over extended time horizons effectively. LSTM has become a foundational model in time-series forecasting, particularly for applications like financial prediction, where long-term dependencies between variables are critical. Its robust design has led to widespread adoption across various domains, including speech recognition, natural language processing, and financial modelling. Notably, its application in financial forecasting has proven effective in capturing sequential dependencies in stock trends, where market behaviours exhibit temporal dynamics and delayed effects. The rigorous theoretical foundation and its empirical success highlight LSTM as a milestone in machine learning. The LSTM model equations are exhibited as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.5}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.7}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.8}$$

$$h_t = o_t * \tanh(C_t), \tag{2.9}$$

where σ and $tanh$ are the activation functions. f_t, i_t and o_t are the forget gate, input gate, and forget gate in LSTM. Other variables are the same implication as RNN. RNN variations are the contemporary SOTA to analyse univariate or multivariate time series data[30, 69, 74]. LSTM-GARCH and LSTM-ARIMA are the most noted hybrid methods in financial data analysis[42, 66]. Nonetheless, Transformer has greater promise in analysing long multivariate sequential data.

2.1.3 AutoEncoder

AutoEncoder (AE) is a long-standing unsupervised neural network traced from 1988 by Bourlard[7]. While in 2006, the paper published by Hinton emblemed that AE could be effectively trained and applied in industry[34]. Afterwards, to solve the over-fit issue in the AE, Vincent added the stochastic noise in the input layer of AE to improve the robustness[72], named Denosing AE. Sparse AE [56], CNN-AE[54], and LSTM-AE[68] are the other well-known variations of AE.

A milestone in AE and Deep Learning development is the inception of Variational AE (VAE)[46], giving a feasible concept of enhancing robustness by combining the probabilistic graph model with the deep learning model. As a generative model, it has fabulous mathematics and gratifying performance, which is still popular in deep financial time series analysis[24]. For the VAE proposed in 2014, we could accentuate the main three components: Encoder, Decoder, and Loss Optimisation. The components of VAE to interpret the significance of VAE in financial applications.

The encoder in VAE is responsible for transforming input data to the latent variable, which is the dimension reduction and an important part of the generative model. The encoder generates a mean vector and a standard deviation vector to construct the multivariate Gaussian distribution in the latent space by encoding MLP. Let the Prior latent variables present $p_\theta(z) = \mathcal{N}(z; 0, I)$. In this case, the variational approximate posterior structure produced by the Encoder is as follows:

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x) I), \tag{2.10}$$

where z is the latent variable, x is the input, ϕ is the parameter of the encoder. The decoder in VAE, which we need to optimise, generally uses Bernoulli MLP (Binary Cross-Entropy) or Gaussian MLP (Mean Squared Error). Superficial expression is:

$$p_\theta(x|z) = \mathrm{Ber}(x; \sigma_\theta(z)) \quad or \quad p_\theta(x|z) = \mathrm{Gau}(x; \sigma_\theta(z)) \tag{2.11}$$

To generate the target x', by the Bayesian equation, it is easily acquired that $x' = \frac{p_\theta(x|z)\cdot p(z)}{p_\phi(z|x)}$. Training the deep learning model to ensure VAE ideally generate the target x', we are demanded to optimise the Loss function of the VAE below:

$$\mathscr{L}(x,\theta,\phi) = -D_{KL}(q_\phi(z|x)||p(z)) + \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] \tag{2.12}$$

Disentangle problems in VAE could be an analogy to signal decomposition in the time series problem. The same focus is on splitting the different characteristics in the original dataset. While the disentangling in VAE is a trending issue, the time-frequency decomposition in the time series data is more well-explored in deep learning applications.

2.1.4 Transformer and Large Language Models

The Transformer model has become the cornerstone of modern deep learning architectures, particularly in natural language processing (NLP). Its popularity surged following the remarkable success of Chat Generative Pre-trained Transformer (ChatGPT) developed by OpenAI, which demonstrated the impressive capabilities of Transformer-based models in generating human-like text and engaging in coherent dialogues [59]. While the attention mechanism, first introduced by Dzmitry Bahdanau in 2014 [3], laid the foundation, the Transformer architecture itself was formally proposed by researchers at Google during the 2017 NeurIPS Conference [71].

The original Transformer paper focused on simplifying the encoder-decoder architecture by removing recurrence and convolution, relying solely on the attention mechanism. The attention mechanism, which is at the core of the Transformer model, processes input sequences in parallel, making it highly efficient and scalable. This innovation marked a departure from traditional recurrent neural network (RNN)-based approaches, allowing for faster training and the ability to capture long-range dependencies in data.

2.1.4.1 Attention Mechanism

The essence of the Transformer lies in its use of the Scaled Dot-Product Attention and Multi-Head Attention mechanisms, which have become fundamental to many deep learning models. Scaled Dot-Product Attention, a local form of attention, is defined as follows:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.13}$$

where Q, K, and V represent the query, key, and value matrices, respectively. These matrices are derived from the input embeddings, which are linearly transformed using learned weight matrices W_Q, W_K, and W_V. The scaling factor $\sqrt{d_k}$ helps stabilize gradients and improve training efficiency. To extend the model's capacity to focus on different aspects of the input sequence, Multi-Head Attention is employed:

$$\begin{aligned} \text{MultiHead}(Q,K,V) &= \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O, \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \end{aligned}$$

Multi-head attention allows the model to learn from different subspaces simultaneously, enhancing its ability to capture complex dependencies.

2.1.4.2 Transformer Variants for Time Series

While the Transformer architecture has revolutionized NLP, it has been adapted and extended to handle other types of data, including time series. Notable variants that have been applied to time series forecasting include the Temporal Fusion Transformer (TFT) [51], which integrates temporal context to enhance predictions, the LogSparse Transformer [50], which optimizes attention by focusing on relevant sparse areas in long sequences, and the Informer [83], which introduces a sparse self-attention mechanism to improve efficiency. Although these models have shown promise in handling sequential data, they have yet to surpass traditional models like LSTM-GARCH in the financial domain, particularly for high-frequency financial data.

2.1.4.3 Large Language Models and Fine-Tuning Techniques

The Transformer's success laid the groundwork for the development of large language models (LLMs), such as GPT-3, GPT-4 [59], and the LLaMA series [70]. These models, pre-trained on vast corpora, have demonstrated exceptional generalization capabilities, often performing well across a wide array of NLP tasks with minimal task-specific fine-tuning. However, various fine-tuning techniques have been developed to adapt these models more effectively to specialized tasks like financial forecasting.

Parameter-Efficient Fine-Tuning: LoRA and QLoRA Low-Rank Adaptation (LoRA) [37] and Quantized LoRA (QLoRA) [20] are advanced techniques designed to fine-tune large language models without the computational overhead associated with traditional fine-tuning methods. LoRA reduces the number of trainable parameters by injecting low-rank matrices into the Transformer layers, making fine-tuning more computationally efficient. This method is particularly advantageous for deploying LLMs on resource-constrained hardware while preserving model performance.

QLoRA extends this concept by leveraging quantization techniques, reducing the model size even further by using lower-precision arithmetic (such as 8-bit or 4-bit integers). This quantization significantly lowers memory usage and allows for faster inference, making it practical to deploy large models in real-time applications, such as financial trading or risk assessment, where speed and efficiency are critical.

Retrieval-Augmented Generation (RAG) RAG [49] is a cutting-edge approach that combines LLMs with retrieval systems to enhance the model's knowledge and generate more accurate and contextually relevant responses. RAG operates by first retrieving relevant documents or passages from a pre-built knowledge base based on the input query. The retrieved information is then used as context for the LLM to generate a response, effectively augmenting the model with external knowledge. In financial forecasting, RAG can be instrumental in leveraging historical financial reports or real-time market data to inform predictions, thereby enhancing the LLM's accuracy and interpretability.

These fine-tuning techniques not only enhance the adaptability of LLMs for specialized tasks but also make it feasible to apply large-scale models in practical financial applications, where data privacy, model interpretability, and computational efficiency are paramount.

2.1.4.4 Implications for Financial Forecasting

The advancements in LLMs and Transformer-based models have opened new avenues for financial forecasting. By incorporating techniques like LoRA, QLoRA, and RAG, these models can be adapted to capture complex financial patterns and respond to rapid market changes. Furthermore, the integration of financial-specific Transformer variants, such as those handling temporal dependencies, promises enhanced accuracy and robustness in financial time series forecasting. As these methods continue to evolve, the potential for Transformer-based architectures in the finance sector will likely expand, offering deeper insights and more reliable predictions.

2.2 Financial Time Series

In financial time series analysis, traditional models have been instrumental in quantifying and predicting relationships within and across various markets. Understanding these foundational models is essential to appreciate how recent deep learning approaches have built upon them to enhance predictive power. This section explores classic financial time series models such as **ARIMA** and **GARCH**, as well as time-frequency models that are widely used in financial data transformations, including Fourier and Wavelet Transformations.

2.2.1 Traditional Time Series Models

Time series models study data points collected over time to identify patterns, trends, and relationships. In financial contexts, time series models are commonly used for applications such as price forecasting, volatility modelling, portfolio optimization, macroeconomic analysis, and algorithmic trading. Two of the most widely used families of traditional financial time series models are **ARIMA** and **GARCH**.

ARIMA Model The *Auto-regressive Integrated Moving Average* (ARIMA) model, proposed by George Box and Gwilym Jenkins in the 1970s [8, 32], integrates both autoregressive (AR) and moving average (MA) components to account for temporal correlations within a time series. The general ARIMA model can be expressed as:

$$(1-\Phi_1 L-\Phi_2 L^2-\cdots-\Phi_p L^p)(1-L)^d X_t=(1+\Theta_1 L+\Theta_2 L^2+\cdots+\Theta_q L^q)\varepsilon_t, \tag{2.14}$$

where X_t is the forecasted time series value, Φ_i are the autoregressive coefficients, Θ_i are the moving average coefficients, L is the lag operator, and ε_t represents the white noise error term. The orders p, d, and q correspond to the autoregressive, differencing, and moving average components, respectively. While ARIMA focuses on predicting the values of a time series, it does not account for volatility, which is addressed by the GARCH model.

GARCH Model The *Generalized Auto-regressive Conditional Heteroskedasticity* (GARCH) model, on the other hand, is designed to model and forecast volatility rather than actual values. GARCH, which extends the *Auto-regressive Conditional Heteroskedasticity* (ARCH) model, is particularly useful for capturing clustering in volatility-a common characteristic of financial time series data [29]. The GARCH model can be represented as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \cdots + \alpha_p \varepsilon_{t-p}^2 + \beta_1 \sigma_{t-1}^2 + \cdots + \beta_q \sigma_{t-q}^2, \tag{2.15}$$

where σ_t^2 is the conditional variance or volatility forecast, ε_t denotes white noise, and α_i and β_i are the coefficients capturing the ARCH and GARCH effects, respectively. The orders p and q represent the number of lags in the ARCH and GARCH terms, respectively.

Hybrid Models With the rise of deep learning, hybrid models that combine traditional statistical methods with neural networks have emerged. For instance, LSTM-GARCH integrates Long Short-Term Memory (LSTM) networks to capture non-linear patterns within a time series, and subsequently uses GARCH to model volatility [42]. Another popular approach is the ARIMA-LSTM model, which leverages ARIMA for linear components and LSTM for non-linear patterns [26].

2.2.2 Factor Models in Financial Analysis

Factor models have been extensively used in finance to explore underlying asset structures and relationships among various financial products. The *Capital Asset Pricing Model* (CAPM) is a foundational single-factor model that describes the relationship between expected returns and market risk. CAPM is expressed as:

$$R_i = R_f + \beta_i (R_m - R_f), \tag{2.16}$$

where R_i is the expected return, R_f is the risk-free rate, β_i represents sensitivity to market risk, and R_m is the expected market return. CAPM has been expanded upon by models such as the *Fama-French Three-Factor Model*, which incorporates additional factors for size and value [25]:

$$R_i = R_f + \alpha_i + \beta_m (R_m - R_f) + \beta_s \mathrm{SMB} + \beta_h \mathrm{HML} + \varepsilon_i, \tag{2.17}$$

where α_i is the asset's excess return, SMB (Small Minus Big) captures size effects, and HML (High Minus Low) captures value effects. Researchers have since extended these models further to incorporate additional factors relevant to financial markets.

2.2.3 Time-Frequency Models for Financial Time Series

Time-frequency methods, particularly Fourier Transform (FT) and Wavelet Transform (WT), play a significant role in financial time series analysis. These methods decompose time series data into constituent frequencies, aiding in the identification of cyclical patterns and volatility structures.

Fourier Transform Fourier Transform is foundational for analyzing cyclic behaviour within financial time series. The Fast Fourier Transform (FFT) [18] is a computationally efficient version of the Discrete Fourier Transform (DFT), widely used for asset pricing and derivative analysis [15]. The FFT can be expressed as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}nk}, \quad k = 0, 1, \ldots, N-1, \tag{2.18}$$

where $X[k]$ is the k-th frequency component, and $x[n]$ is the n-th time sample. However, FFT assumes stationary data, which limits its applicability to financial time series that often exhibit non-stationary behavior.

Wavelet Transform Wavelet Transform (WT) addresses the non-stationary nature of financial data more effectively than FT by using wavelets to analyze both time and frequency simultaneously. The Discrete Wavelet Transform (DWT) can be defined as:

$$d_{j,k} = \sum_{n=0}^{N-1} f[n]\psi_{j,k}(n), \tag{2.19}$$

where $\psi_{j,k}(n)$ is the wavelet function at scale j and translation k, providing multi-resolution analysis [53]. WT is particularly valuable for capturing localized time-frequency characteristics within financial time series.

Empirical Mode Decomposition (EMD) Empirical Mode Decomposition (EMD) is another decomposition technique used for non-linear, non-stationary time series [38]. EMD decomposes a signal into Intrinsic Mode Functions (IMFs), each representing distinct oscillatory modes:

$$f(n) = \sum_{m=1}^{M} IMF_m(n) + Res_M(n), \tag{2.20}$$

where $IMF_m(n)$ represents the m-th oscillatory mode and $Res_M(n)$ is the residual. EMD's ability to capture complex behaviours in financial time series makes it suitable for multi-scale analysis. Recent advancements, such as Variational Mode Decomposition (VMD) [23], further enhance decomposition by addressing issues like mode mixing.

2.2.4 Hybrid Models with Deep Learning

In recent years, hybrid models combining traditional time-frequency methods with deep learning architectures have gained traction. These models apply EMD or VMD to decompose financial time series into components that can be individually modelled using neural networks [86]. While this improves forecasting accuracy by isolating different frequency components, it can also introduce complexity due to the increased dimensionality when handling multi-variate series.

These advanced techniques, combined with deep learning models, continue to push the boundaries of financial time series forecasting, allowing for more nuanced and accurate predictions. However, they also highlight challenges in integrating traditional statistical models with modern machine learning approaches, particularly in the context of financial data's unique properties.

2.3 Evaluation Metrics

In evaluating the performance of financial time series forecasting models, several key metrics are commonly used to quantify accuracy and predictive quality. This section describes the metrics employed in this study, including Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Information Coefficient (IC), Rank Information Coefficient (Rank IC), Information Coefficient Information Ratio (ICIR), and Rank Information Coefficient Information Ratio (Rank ICIR).

2.3.1 Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) measures the average magnitude of errors between predicted and actual values, without considering their direction. It is defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{2.21}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the total number of predictions. MAE provides an intuitive measure of accuracy, indicating how close the predicted values are to the true values on average.

2.3.2 Mean Squared Error (MSE)

The Mean Squared Error (MSE) measures the average squared difference between predicted and actual values, penalizing larger errors more heavily than MAE. MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{2.22}$$

Due to its quadratic nature, MSE is more sensitive to outliers, making it a useful metric when large prediction errors are particularly undesirable.

2.3.3 Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) measures the average percentage difference between predicted and actual values, which normalizes the error by the actual value. It is defined as:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%. \tag{2.23}$$

MAPE is particularly useful when comparing model performance across datasets of varying scales. However, it can be biased by values close to zero, as the percentage error becomes exaggerated.

2.3.4 Information Coefficient (IC)

The Information Coefficient (IC) measures the correlation between predicted and actual returns. It reflects how well the predictions align with actual market movements. IC is defined as the Pearson correlation coefficient between the predictions and actual values:

$$\text{IC} = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}, \tag{2.24}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the mean values of the actual and predicted returns, respectively.

2.3.5 Rank Information Coefficient (Rank IC)

Rank Information Coefficient (Rank IC) is a variant of IC that uses the rank values of predicted and actual returns rather than their raw values. It is defined as the Spearman rank correlation between the predicted ranks and actual ranks:

$$\text{Rank IC} = \frac{\sum_{i=1}^{n}(\text{rank}(y_i) - \overline{\text{rank}(y)})(\text{rank}(\hat{y}_i) - \overline{\text{rank}(\hat{y})})}{\sqrt{\sum_{i=1}^{n}(\text{rank}(y_i) - \overline{\text{rank}(y)})^2 \sum_{i=1}^{n}(\text{rank}(\hat{y}_i) - \overline{\text{rank}(\hat{y})})^2}}. \tag{2.25}$$

2.3.6 Information Coefficient Information Ratio (ICIR)

The Information Coefficient Information Ratio (ICIR) evaluates the consistency of IC over time. It is calculated by dividing the mean of the ICs by their standard deviation:

$$\text{ICIR} = \frac{\mathbb{E}[\text{IC}]}{\sigma_{\text{IC}}}, \tag{2.26}$$

where $\mathbb{E}[\text{IC}]$ is the average IC over a period, and σ_{IC} is the standard deviation of ICs over the same period.

2.3.7 Rank Information Coefficient Information Ratio (Rank ICIR)

Rank ICIR is the consistency measure for Rank IC over time, computed as:

$$\text{Rank ICIR} = \frac{\mathbb{E}[\text{Rank IC}]}{\sigma_{\text{Rank IC}}}, \tag{2.27}$$

where $\mathbb{E}[\text{Rank IC}]$ is the mean of the Rank IC values over a given period, and $\sigma_{\text{Rank IC}}$ is the standard deviation of these Rank IC values.

2.3.8 Relationship to Loss Function

The loss function used in training the model directly influences the model's performance as measured by these evaluation metrics. Typically, the choice of loss function is aligned with one of the metrics. For example, models trained using Mean Squared Error (MSE) loss are optimized to minimize squared errors and may perform well on MSE and MAE metrics. The FinLoss metric in this thesis is specifically designed to optimise the Information Coefficient (IC), which directly influences IC, ICIR, Rank IC, and Rank ICIR metrics. These metrics measure the model's effectiveness in capturing relationships within financial time series data, which aligns with the objectives of financial forecasting tasks.

CHAPTER 3

Chapter 3 Literature Review

The study of financial time series forecasting has been a prominent area of research within both the statistical and data science communities for several decades [19]. This chapter begins by exploring the significant advancements in deep learning methodologies, which have been increasingly applied to model complex dependencies in financial data. Unlike traditional statistical models, deep learning approaches-such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformers-excel at capturing the intricate, non-linear patterns prevalent in financial markets.

The discussion then transitions to the evolution of these deep learning models, demonstrating how their architectures have advanced from foundational neural networks to more sophisticated frameworks capable of handling large-scale financial datasets. This review also highlights specific applications of deep learning in forecasting financial time series, such as predicting stock prices, modelling volatility, and assessing market risk. Finally, the chapter provides an overview of the current challenges and opportunities in the field, emphasizing the transformative impact of deep learning techniques on improving predictive accuracy and model robustness in financial forecasting.

3.1 Deep Time Series Forecasting in Finance

Deep learning methods have transformed financial time series forecasting by offering advanced techniques to model temporal and relational dependencies in complex datasets. This section explores key advancements, including foundational architectures like LSTM, enhanced frameworks such as ALSTM, and novel methods like HIST and multi-order dynamic integration. These approaches demonstrate how deep learning has addressed challenges such as data noisiness, non-stationarity, and the intricate relationships between financial entities, paving the way for more accurate and robust predictive models in finance.

3.1.1 Enhancing Stock Movement Prediction with Adversarial Training

Feng et al. [27] proposed a novel application of adversarial training to enhance stock movement prediction. The authors developed an Adversarial Long Short-Term Memory (ALSTM) model that combines traditional LSTM capabilities with adversarial noise injection to improve model robustness. The approach leverages adversarial examples to augment the training process, enabling the model to withstand perturbations and noise typical of real-world financial data. Unlike traditional LSTMs, ALSTM introduces a new adversarial loss component that regularizes the model and improves generalization performance. The model demonstrated superior predictive accuracy on multiple benchmark datasets, including financial indices and sector-specific stocks, compared to traditional LSTMs and baseline models. This paper is significant because it addresses one of the most challenging aspects of financial prediction-data noisiness and non-stationarity. By integrating adversarial training, ALSTM provides a promising direction for developing more resilient predictive financial systems, with potential applications in trading strategies and risk management.

3.1.2 Temporal Relational Ranking for Stock Prediction

Temporal Relational Ranking (RSR) methods focus on capturing temporal and relational dependencies between financial assets to enhance stock prediction [39]. RSR emphasizes the role of interdependencies among financial entities by combining time-series data with relational dynamics, making it possible to model interactions between stocks influenced by economic sectors, supply chains, or shared market factors. These models often employ advanced neural architectures or ranking mechanisms to prioritize meaningful temporal features, such as historical price trends, volatility, and trading volumes. Recent research has explored hybrid approaches combining RSR with graph-based methods, creating frameworks capable of learning temporal signals and structural dependencies. Although specific implementation details may vary, RSR's relevance lies in its potential to bridge the gap between temporal modeling (e.g., LSTM) and relational reasoning (e.g., graph neural networks). Future work could expand on integrating RSR with attention mechanisms or transformer models to capture intricate relationships better.

3.1.3 HIST: A Graph-Based Framework for Stock Trend Forecasting

Xu et al. [78] introduced HIST, a graph-based framework designed to improve stock trend forecasting by mining concept-oriented shared information. The method leverages graph neural networks to model the relationships between stocks, capturing local dependencies (e.g., sector correlations) and global (e.g., macroeconomic influences). By incorporating shared information across related financial instruments, HIST aims to reduce noise and enhance the interpretability of predictions. The framework introduces a novel concept-sharing mechanism, enabling more robust feature extraction and the identification of hidden patterns in stock behaviours. HIST demonstrated superior performance in multiple evaluation scenarios, achieving higher accuracy and lower error rates than traditional LSTM-based models. This work is particularly notable for its innovative use of graph-based representations in financial forecasting, providing a new paradigm for modelling complex relationships in stock markets. HIST's ability to mine concept-oriented features highlights its potential for broader applications like portfolio optimization and anomaly detection.

3.1.4 Efficient Integration of Multi-Order Dynamics in Stock Movement Prediction

Huynh et al. [39] presented an efficient framework for stock movement prediction that integrates multi-order relational dynamics and internal temporal dynamics. Published in WSDM 2023, this work addresses the challenge of capturing complex, hierarchical dependencies in financial data. The proposed framework employs a multi-order relational model to capture first-order interactions (direct relationships between stocks) and higher-order interactions (indirect relationships mediated through other entities). Additionally, the model incorporates internal temporal dynamics to account for time-specific trends and seasonality. The authors demonstrated the scalability and predictive power of the framework on large-scale financial datasets, including global stock indices and cryptocurrency markets. This research sets a benchmark for future work in scalable financial forecasting methods by balancing computational efficiency with predictive accuracy. Integrating multi-order dynamics introduces significant potential for applications in high-frequency trading and algorithmic decision-making.

3.2 Large Language Model in Time Series Forecasting

Applying Large Language Models (LLMs) in time series forecasting has emerged as a transformative approach, leveraging their powerful sequence modelling and reasoning capabilities. By reprogramming and fine-tuning these models to handle temporal data, researchers have demonstrated their ability to capture complex patterns, dependencies, and relationships critical for accurate predictions. Recent advancements have focused on integrating multimodal inputs, aligning textual and temporal representations, and enhancing anomaly detection through prompt engineering and specialized fine-tuning techniques. These innovations highlight the versatility of LLMs in adapting to diverse time series forecasting tasks, paving the way for more robust and generalizable solutions in domains such as finance, healthcare, and climate modelling.

3.2.1 AnyGPT: Unified Multimodal LLM with Discrete Sequence Modeling

Zhan et al. [82] introduced AnyGPT, an innovative multimodal language model that processes diverse data types-such as speech, text, images, and music-through discrete representations. This approach enables the seamless integration of new modalities into existing LLM architectures without necessitating structural modifications. By employing data-level preprocessing, AnyGPT treats the incorporation of new modalities similarly to adding new languages, simplifying the training process. The authors constructed a multimodal text-centric dataset for alignment pre-training and synthesized a large-scale multimodal instruction dataset comprising 108,000 samples of multi-turn conversations. Experimental results demonstrated that AnyGPT effectively facilitates any-to-any multimodal conversations, achieving performance on par with specialized models across various modalities. This study underscores the potential of discrete representations in unifying multiple modalities within a language model, offering a versatile framework for multimodal understanding and generation.

3.2.2 Can LLMs Serve As Time Series Anomaly Detectors?

Dong, Huang, and Cao [22] investigated the applicability of LLMs, specifically GPT-4 and LLaMA3, in detecting and explaining anomalies within time series data-a critical task in numerous real-world applications. Their findings revealed that LLMs, in their standard form, are not directly suitable for time series anomaly detection. However, by implementing prompt strategies such as in-context learning and chain-of-thought prompting, GPT-4 exhibited competitive performance relative to baseline methods. To further enhance detection capabilities, the authors developed a synthesized dataset that automatically generates time series anomalies accompanied by corresponding explanations. Instruction fine-tuning on this dataset led to improved performance of LLaMA3 in anomaly detection tasks. This study highlights the promising potential of LLMs as time series anomaly detectors, provided that appropriate prompt engineering and fine-tuning techniques are applied.

3.2.3 TIME-LLM: Time Series Forecasting by Reprogramming Large Language Models

Jin et al. [40] presented TIME-LLM, a framework that repurposes LLMs for general time series forecasting without altering the underlying language model architecture. The authors reprogrammed input time series data into text prototypes compatible with LLMs, facilitating alignment between time series and natural language modalities. To enhance the model's reasoning capabilities with time series data, they introduced the Prompt-as-Prefix (PaP) technique, enriching input context and guiding the transformation of reprogrammed input patches. The transformed patches were then projected to generate forecasts. Comprehensive evaluations demonstrated that TIME-LLM outperformed state-of-the-art specialized forecasting models and exhibited strong performance in both few-shot and zero-shot learning scenarios. This work illustrates the adaptability of LLMs to time series forecasting tasks through strategic reprogramming and prompt engineering.

3.2.4 CALF: Aligning LLMs for Time Series Forecasting via Cross-Modal Fine-Tuning

Liu et al. [52] proposed CALF, a framework aimed at aligning LLMs for multivariate time series forecasting by addressing the distribution discrepancies between textual and temporal input tokens. The framework comprises two branches: a temporal target branch processing time series input and a textual source branch handling aligned textual input. To reduce distribution discrepancies, the authors developed a cross-modal match module for input alignment, a feature regularization loss to align intermediate features, and an output consistency loss to ensure corresponding output representations. CALF demonstrated state-of-the-art performance in both long-term and short-term forecasting tasks, exhibiting favorable few-shot and zero-shot capabilities. This study emphasizes the importance of modality alignment in fine-tuning LLMs for time series forecasting, offering a robust approach to enhance predictive accuracy.

Part II

Part II

CHAPTER 4

Fourier Graph Convolution Transformer for Financial Multivariate Time Series Forecasting

Financial Multivariate Time Series (Fin-MTS) forecasting is increasingly critical in the financial market. Unlike other Multivariate Time Series (MTS) data, Fin-MTS exhibits particular characteristics, including non-linearity, volatility, and hidden periodicities, which thus introduce great challenges for modelling it well. Existing state-of-the-art models for Fin-MTS forecasting often overlook hidden periodic characteristics, such as credit and monetary policy cycles. More importantly, these models usually show limited capability in well capturing the intra-series and inter-series dynamic information during the modelling process, resulting in significant information loss in quantitative finance modelling and thus limited forecasting performance. To this end, in this chapter, we introduce a novel model called *Fourier Graph Convolution Transformer* (FreTransformer) for Fin-MTS modelling and forecasting. FreTransformer is not only able to well model both the intra- and inter-series dynamic dependencies, but also well capture the important hidden periodicities embedded in Fin-MTS data. FreTransformer first maps the original time domain data into the frequency domain to disclose the hidden periodicities and then employs a novel Fourier Graph Convolution Network to well capture the complex intra- and inter-series dependencies within Fin-MTS. Extensive experiments on real-world US market data across 12 phases demonstrate that our method outperforms current state-of-the-art models. Our source code is publicly available at this repository: `https://github.com/AmsonntagChow/FreTransformer`

4.1 Introduction

Along with the rapid development of Artificial Intelligence (AI) and its wide applications, AI models provide promising possibilities for Multivariate Time Series (MTS) forecasting in various application scenarios in the real world, such as traffic flow forecasting, financial market prediction [13, 14, 80]. Actually, MTS forecasting in finance has been a critical research problem for a long time. Conventional quantitative finance analysis methods combine mathematical and statistical techniques and financial theories to analyse financial assets, attaining heightened attention in the past decades [8, 35]. In recent years, benefiting from the power of advanced AI, advanced quantitative financial models have exhibited superior performance, both experimental and theoretical. These models generally provide accurate forecasting by capturing the high dimensional, heterogeneous, non-stable, non-i.i.d dependencies within and across Financial Multivariate Time Series (Fin-MTS) data [39]. They have been widely utilised by global-known finance organizations including Hedge Funds, Renaissance Technologies and Two Sigma. These AI models have been able to help these organizations earn more excess returns due to the deep and robust modelling of various complex dependencies embedded in finance data.

In the early years, the modelling of financial market movements relied on linear relationships over historical finance data, e.g., historical asset price data. Linear regression methods based on technical indicators, such as Moving Average (MA) [32] and Auto-Regression (AR) [8], are widely applied among professional and amateur finance traders. However, in the real world, the financial time series data often exhibits *non-linearity and hidden periodicities* in the real world, such as the credit cycle and monetary policy cycle. These characteristics prevent the aforementioned linear models from achieving perfect performance.

Therefore, in recent years, advanced deep learning approaches, including basic recurrent neural network (RNN) and long-short-term memory networks (LSTM), have been introduced to better model the financial time series data. These models have shown promising capability to capture the nonlinear relationships embedded in financial time series data, especially the intra-series relationships. To capture the inter-series relationships which commonly exist in Fin-MTS data, more advanced financial deep learning models, including RSR [28], HIST [78] and ESTIMATE [39] have been developed. These models generally utilise Graph Neural Networks (GNNs) to capture inter-series relationships effectively. For instance, RSR is a framework for stock MTS forecasting that integrates Temporal Graph Convolution and LSTM to learn complex relationships in stock data with a focus on intra-series dynamics. Although promising performance has been achieved, there are still two significant gaps which prevent the further improvement of the performance of these existing methods on Fin-MTS data. On the one hand, there is a lack of a unified framework that can effectively capture both the intra- and inter-time series decencies simultaneously and integrate them well. On the other hand, most of the existing methods cannot effectively capture the hidden periodicities which is commonly embedded in Fin-MTS data, which is significant for accurate Fin-MTS forecasting.

In order to bridge the above significant gaps, in this paper, we introduce a novel model, called Fourier Graph Convolution Transformer (FreTransformer), for Fin-MTS forecasting. To be specific, FreTransformer first maps the original Fin-MTS data into a new latent space, i.e., frequency domain, and then introduces a novel Fourier Graph Convolution Network (FGCN) enhanced Transformer structure to learn both the intra- and inter-series dependencies in a unified way. In particular, the FGCN is able to effectively learn the weight matrices that encode both the intra- and inter-series dependencies in the frequency domain. At the same time, after the transformation from the original data space to the frequency domain, the hidden periodicities is well disclosed and thus can be easily captured in the frequency domain. Thanks to such a novel design, our proposed FreTransformer is able to effectively learn both the intra- and inter-series dependencies in a unified way while emphasising the hidden periodicities. The main contributions of this work can be summarised below:

- We propose a novel framework called FreTrasnformer to effectively capture both the intra- and inter-series dependencies as well as hidden periodicities for Fin-MTS forecasting.

- In FreTransformer, Fourier transform is introduced to map the original data into the frequency domain to effectively disclose the hidden periodicities, which is very hard to directly capture from the original data.

- In addition, a novel FGCN-enhanced transformer is proposed to learn both intra- and inter-series dependencies in the frequency domain in a unified way and naturally integrate them in a learnable complex value matrix.

Extensive experiments on real-world financial data from Yahoo Finance were conducted to assess the performance of FreTransformer in comparison with representative and/or state-of-the-art Fin-MTS forecasting models including LSTM, ALSTM, RSR, HIST, and ESTIMATE. Experimental results demonstrate that FreTransformer outperforms the five baseline models and the rationality of its design.

4.2 Issues and Related Work

Here, we briefly introduce the background of FTS forecasting models by focusing on their objectives and mechanisms to address the financial dataset's intrinsic features. These are mostly relevant to the target of our work. In particular, FTS anomaly detection will be introduced in the following section. Other developments on FTS tasks, such as defining FTS correlation[11], Fig. 4.1, FTS trading, and FTS factors classification are excluded due to their irrelevance to this work.

4.3 Fin-MTS Forecasting and Challenges

Given a Financial Multivariate Time Series (Fin-MTS) input, $\boldsymbol{X_t} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_t}]^T \in \mathbb{R}^{T \times N}$, where $\boldsymbol{x_t} \in \mathbb{R}^N$ represents the N features at time t. The lookback window $\boldsymbol{\hat{X}_t} = [\boldsymbol{x_{t-T+1}}, \ldots, \boldsymbol{x_t}]^T \in \mathbb{R}^{T \times N}$ as the lag correlation feature. Fin-MTS forecasting problem can be defined as predicting the value of next τ time $\boldsymbol{Y_t} = [\boldsymbol{x_{t+1}}, \ldots, \boldsymbol{x_{t+\tau}}] \in \mathbb{R}^{\tau \times N}$ grounded in the historical $\hat{t}$ time observations $\boldsymbol{\hat{X}_{\hat{t}}} = \left[\boldsymbol{x_{\hat{t}-T+1}}, \ldots, \boldsymbol{x_{\hat{t}}}\right]^T \in \mathbb{R}^{\hat{t} \times N}$. Formulas lead to the Fin-MTS forecasting procedure as follows:

$$\boldsymbol{\hat{Y}_t} = \boldsymbol{f}_\theta(\boldsymbol{\hat{X}_t}) = \boldsymbol{f}_\theta\,[\boldsymbol{x_{t-T+1}}, \ldots, \boldsymbol{x_t}]^T \in \mathbb{R}^{\hat{\tau} \times N}, \tag{4.1}$$

where $\boldsymbol{\hat{Y}_t}$ denotes the entire forecast series based on the ground truth $\boldsymbol{X_t}$ for $\hat{\tau}$ future time and $\boldsymbol{f}_\theta$ is the forecasting function with parameters θ.

However, the Fin-MTS forecasting problem has several challenges: Non-stationarity of financial time series data makes model training complex, as the assumption that past patterns predict the future is less reliable [13]. The inherent high volatility due to numerous influencing factors adds to the prediction difficulty. The selection and engineering of predictive features (N dimensions in $\boldsymbol{x_t}$) are important tasks that significantly impact forecasting accuracy. The model complexity required to capture financial data dynamics often leads to overfitting, where models outperform training data but perform poorly on the backtesting data. Moreover, quantifying the uncertainty of predictions is critical yet challenging in financial applications. This uncertainty is more effectively addressed in Fourier Space, where the time series is analyzed through frequencies and amplitudes. It helps isolate and understand the components driving uncertainty, offering a more

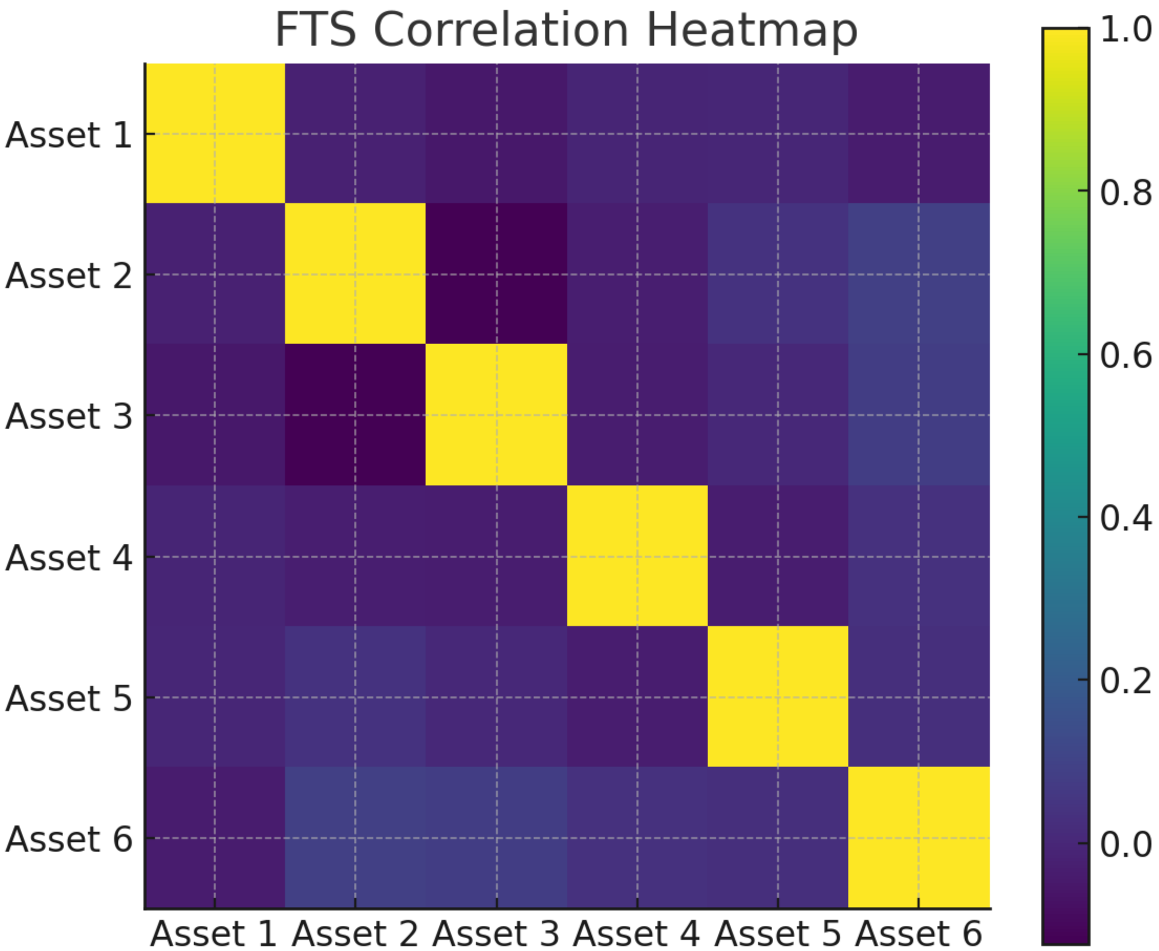


Figure 4.1: Asset-level correlation matrix illustrating the FTS correlation.

straightforward path to managing it. Additionally, sudden events in financial markets can dramatically change underlying data patterns, requiring models to be adaptable and responsive. For example, the Black Swain Events, COVID-19, totally accelerated the changing of the global economic pattern. Financial time series exhibit both long and short-term dependencies, requiring advanced modelling techniques such as LSTM, which adopts the gates technique to effectively capture both long and short-term dependencies.

Addressing these challenges demands a comprehensive understanding of advanced statistical methods, deep learning techniques, robust data preprocessing, and a deep understanding of financial markets. Approaches such as feature selection algorithms, regularization, ensemble models, and advanced neural architectures play crucial roles in overcoming these research and industry challenges. Furthermore, integrating external data sources and adopting adaptive learning strategies are vital for enhancing model sensitivity to market conditions, significantly resulting in more accurate and reliable Fin-MTS forecasts.

4.4 FTS Forecasting Models and Gaps

Fin-MTS has adopted Graph Neural Networks (GNN) because of their superior performance in modelling the complex structural representations among the different assets[39, 43, 63, 65, 78]. Most GNNs adopt a pre-fixed graph structure to capture the correlations and representations among assets, as exemplified by models like HIST[78]. Some models, such as MAN-SF[64] and TRACER[16], have successfully employed attention mechanisms to learn cross-asset correlations in graphs without explicit domain knowledge. Specifically, MAN-SF utilises a graph attention network to learn latent representations, while TRACER uses a concatenated attention mechanism to integrate varying weights of intra-series relationships. Nevertheless, these GNN approaches consistently implement the graph network to study the intra- and inter-series features separately[39, 64, 78] while in the frequency domain is barely feasible due to the intra- and inter-series information is in different representation when the data is changed to harmonic signals. Therefore, the original intra- and inter-series information changes in the frequency domain. In our study, we introduce a frequency-based Transformer with the Fourier Graph Convolution Network (FGCN), which is inspired by the FourierGNN[81] to conduct a novel study of capturing intra- and inter-series information within the Fourier Space harmonic signals.

Although promising performance has been achieved, there are still two significant gaps that prevent further improvement of the performance of these existing methods on Fin-MTS data. On the one hand, there is a lack of a unified framework that can effectively capture both the intra- and inter-time series decencies simultaneously and integrate them well.

4.5 FreTransformer

4.5.1 Introduction

In order to bridge the above significant gaps, in this CA1 report, we introduce a novel model, called Fourier Graph Convolution Transformer (FreTransformer), for Fin-MTS forecasting. To be specific, FreTransformer first maps the original Fin-MTS data into a new latent space, i.e., frequency domain, and then introduces a novel Fourier Graph Convolution Network (FGCN) enhanced Transformer structure to learn both the intra- and inter-series dependencies in a unified way. In particular, the FGCN is able to effectively learn the weight matrices that encode both the intra- and inter-series dependencies in the frequency domain. At the same time, after the transformation from the original data space to the frequency domain, the hidden periodicities is well disclosed and thus can be easily captured in the frequency domain. Thanks to such a novel design, our proposed FreTransformer is able to effectively learn both the intra- and inter-series dependencies in a unified way while emphasising the hidden periodicities.

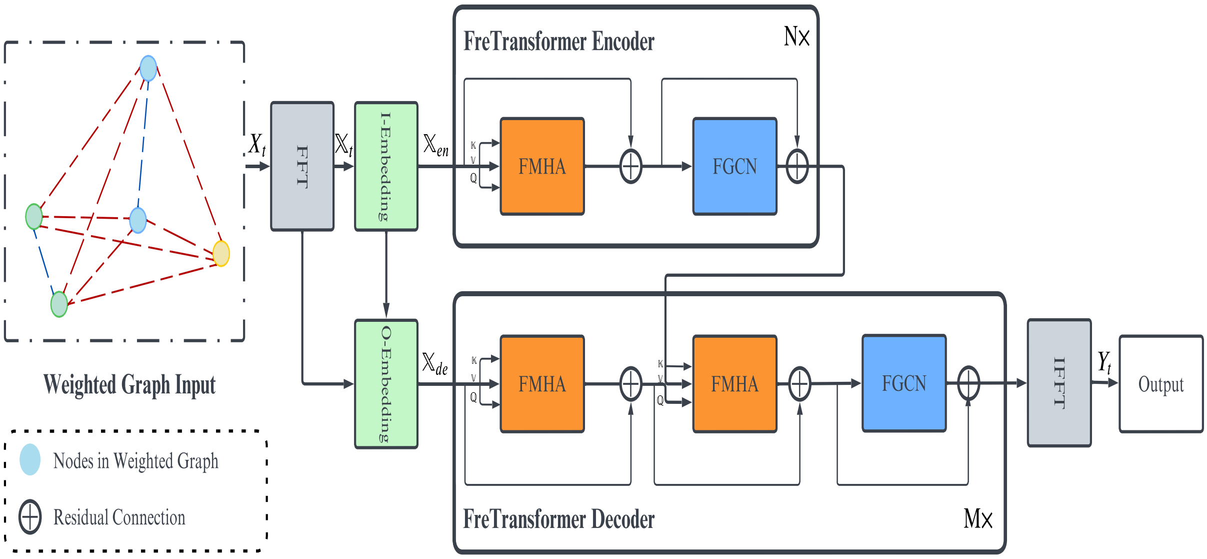


Figure 4.2: FreTransformer Architecture. $\mathbb{X}_t$ discloses the hidden periodicities. The FGCN effectively captures both the intra- and inter-series dependencies as well as hidden periodicities. The Fourier Multi-Head Attention Mechanism (FHMA) is a naïve variant of the basic Multi-Head Attention Mechanism.

4.5.2 Frouier Transform

To effectively analyse and interpret the underlying patterns within financial multivariate time series (Fin-MTS), it's crucial to transition from analysing these phenomena in the time domain to a more revealing frequency domain perspective. The Fourier Transform provides us with this exact capability, serving as a cornerstone analytical tool that is both fundamental and lossless. It enables us to unveil the cyclical and periodic nature of financial data, which might be obscured when viewed strictly in the time sequence. The transition to the frequency domain offers a clearer understanding of the inherent dynamics and trends within the data, paving the way for more informed decision-making and predictive modelling. Given the discrete nature of Fin-MTS data, we employ the Fast Fourier Transform (FFT) [18] to efficiently decompose our time series data. This process not only preserves the information contained within the original time series but also highlights cyclical trends by converting the discrete input, denoted as $\boldsymbol{X_t}$, into its frequency domain counterpart. This transformation is instrumental in capturing the underlying cyclical patterns within the data, thus offering a significant advantage in our analysis and understanding of financial time series. FFT is shown as:

$$\begin{aligned}
\mathbb{X}_t &= \mathrm{FFT}(\boldsymbol{X_t}) = A(\boldsymbol{X_t})e^{j\Phi(\boldsymbol{X_t})} \\
&= \left[\sum_{n=0}^{N-1} X_t[t,n]\cdot e^{-\frac{2\pi i}{N}kn}\right]_{k=0}^{N-1} \\
&= \mathrm{Re}\left(\sum_{n=0}^{N-1} X_t[t,n]\cdot e^{-\frac{2\pi i}{N}kn}\right) \\
&+ i\cdot \mathrm{Im}\left(\sum_{n=0}^{N-1} X_t[t,n]\cdot e^{-\frac{2\pi i}{N}kn}\right), \\
&\forall t \in \{1,2,\ldots,T\},
\end{aligned} \tag{4.2}$$

where we have $\mathbb{X}_t \in \mathbb{C}^{\frac{T}{2}\times N}$ as FFT output, denoting the Fin-MTS data in the frequency domain. $A(\boldsymbol{X_t})$ is the amplitude and $\Phi(\boldsymbol{X_t})$ is the phase. To briefly present in this study, we have the abbreviation $\mathbb{X}_t = Re(\mathbb{X}_t) + i\cdot Im(\mathbb{X}_t)$, where $Re(\mathbb{X}_t)$ and $Im(\mathbb{X}_t)$ separately rewrite the real part and imaginary part in the Equation 4.2.In subsequent sections, for simplicity, we will refer to Equation 4.2 as `FFT` and Equation 4.3 as `iFFT`. As one of the most critical recent algorithms, FFT has the lossless reversal method iFFT[18], which is shown as:

$$\begin{aligned}
\boldsymbol{X_t} &= \mathrm{IFFT}(\mathbb{X}_t) \\
&= \left[\frac{1}{N}\sum_{k=0}^{N-1} \mathbb{X}_t[t,k]\cdot e^{\frac{2\pi i}{N}kn}\right]_{n=0}^{N-1} \\
&= \frac{1}{N}\left(\mathrm{Re}\left(\sum_{k=0}^{N-1} \mathbb{X}_t[t,k]\cdot e^{\frac{2\pi i}{N}kn}\right)\right) \\
&+ i\cdot \frac{1}{N}\left(\mathrm{Im}\left(\sum_{k=0}^{N-1} \mathbb{X}_t[t,k]\cdot e^{\frac{2\pi i}{N}kn}\right)\right), \\
&\forall t \in \{1,2,\ldots,T\}.
\end{aligned} \tag{4.3}$$

4.5.3 Prior Fully Connected Weighted Graph

Then, we introduce the prior graph structure, a novel approach in Fin-MTS forecasting scenarios[4, 9, 77, 81] leads to the following definitions for forming a Fully Connected Weighted Graph. The graph is shown below as the Fig 1.1. After transforming to a weighted graph, the original task from Eq. 4.2 can be reformulated as:

$$\hat{\boldsymbol{Y}}_{\boldsymbol{t}} = \boldsymbol{f}_{\theta,\theta_g}(\hat{\boldsymbol{X}}_{\boldsymbol{t}}) = \boldsymbol{f}_{\theta,\theta_g}[\boldsymbol{x}_{\boldsymbol{t-T}+\boldsymbol{1}},\ldots,\boldsymbol{x}_{\boldsymbol{t}}]^T \in \mathbb{R}^{\hat{\tau}\times N}. \tag{4.4}$$

Equation 4.4 is a rewritten formula with the parameter θ_g, which means the graph network learns the intra- and inter-series features representation with θ_g represents the FGCN hyperparameter and other parameters θ.

4.5.4 Fourier Graph Convolution Network

To better capture the intra- and inter-series features in the Fin-MTS data, we propose an enhanced module in the Fourier domain FGCN based on the FourierGNN[81].

Proposition 1 (Fourier Graph Convolution Network).
For a weighted graph $G_W = (X, A)$ with the $X \in \mathbb{R}^{T\times N}$ as the nodes and the $A \in \mathbb{R}^{T\times T}$ as the adjacency matrix to study the inner series representation, where the T is the number of nodes (time) and N as the features. We have a learnable weight matrix $W \in \mathbb{R}^{N\times N}$ to learn the cross-series information. By applying Hadamard product on A and W, we can form a tailored Green's kernel $\kappa : [N] \times [N] \to \mathbb{R}^{T\times T}, T > N$ with $\kappa[i,j] := A_{ij} \circ W$.

The difference between traditional convolution neural network (CNN)[48] compared to the pure math convolution concept is only the kernel direction, where the pure math convolution has a derivative formula in the frequency domain based on the convolution theorem[41] can be rewritten as:

$$\mathrm{FFT}(X)\mathrm{FFT}(\kappa) = \mathrm{FFT}((X * \kappa)[i][j]), \tag{4.5}$$

where for all $i \in [N]$ and $j \in [N]$. We can implement the weighted graph concept $G_W = (X, A)$ to constitute the FGCN from Equation 4.2, 4.3 and 4.5 as:

$$\begin{gathered}
\mathrm{FGCN}_W(X, A) = \sigma\Big(\sum_{l=0}^{L}(\mathrm{FFT}(AXW) + \mathbb{B})\Big) \\
\mathrm{FGCN}_{\mathfrak{K}}(\mathbb{X}) = \sigma\Big(\sum_{l=0}^{L}(\mathbb{X} \times \mathfrak{K}[i][j] + \mathbb{B})\Big).
\end{gathered} \tag{4.6}$$

Proof. The proof demonstrates the equivalence in Equation 4.6 between the complex input in Fourier space and graph input in the time domain. It also confirms that the unified complex learnable weighted matrix $\mathfrak{K}$ contains the same information learned from the weighted matrices W and A. According to $\kappa[i,j] := A_{ij} \circ W$, we can expand the graph convolutions AXW in the time domain to Fourier space:

$$\begin{aligned}\mathrm{FFT}(AXW) &= \mathrm{FFT}\left(\sum_{i=1}^{n} A_{ij}XW[j]\right) = \mathrm{FFT}\left(\sum_{i=1}^{n} X[j]k[i,j]\right)\\ &= \mathrm{FFT}((X * \kappa)[i][j]) = \mathrm{FFT}(X)\mathrm{FFT}(\kappa[i][j])\\ &= \mathbb{X} \times \mathfrak{K}[i][j],\end{aligned}$$ where

we arrive at the equivalence expression $\mathrm{FGCN}_W(X,A) = \mathrm{FGCN}_{\mathfrak{K}}(\mathbb{X})$ in the Equation 4.6. ■

Thus, we can incorporate the FGCN into the experimental programming framework as detailed in Section III-D by reformulating it as the following equation:

$$\begin{aligned}&\mathbb{X} \times \mathfrak{K}[i][j] + \mathbb{B}\\ =&(Re(\mathbb{X}) + i \cdot Im(\mathbb{X}))\cdot\\ &(Re(\mathfrak{K})[i][j] + i \cdot Im(\mathfrak{K}[i][j])) + \mathbb{B}\\ =&(Re(\mathbb{X})Re(\mathfrak{K})[i][j] - Im(\mathbb{X})Im(\mathfrak{K}[i][j]) + Re(\mathbb{B}))+\\ &i(Re(\mathbb{X})Im(\mathfrak{K}[i][j]) + Im(\mathbb{X})Re(\mathfrak{K})[i][j] + Im(\mathbb{B})).\end{aligned} \tag{4.7}$$

4.5.5 FreTransformer

We have innovated the Transformer[71] model by integrating it with an FGCN, creating a frequency-based deep learning structure named FreTransformer, as shown in Fig. 4.2. This structure includes the Weighted Graph FFT Input along with a corresponding Encoder and Decoder.

FreTransformer Input. The inputs of the encoder and decoder part are denoted as $\mathbb{X}_{en} \in \mathbb{C}^{\frac{T}{2} \times N}$ and $\mathbb{X}_{de} \in \mathbb{C}^{\frac{T}{2} \times N}$. Each embedding initialisation is combined with two parts: FGCN Fin-MTS embedding to learn the representation of the assets data and the positional embedding (PE) [71] to learn the intra-series features with phase in the frequency domain. The inputs are formulated as follows:

$$\begin{aligned}\mathbb{X}_{en} &= \mathrm{FGCN}_{\mathfrak{K}}(\mathbb{X}) + \mathrm{PE}(\mathbb{X})\\ \mathbb{X}_{de} &= \mathrm{Concat}(\mathbb{X}_{en}, \mathbb{X}).\end{aligned} \tag{4.8}$$

Fourier Multi-Head Attention Mechanism. The Fourier Multi-Head Attention (FMHA) is a variation of the original attention mechanism[71]. We directly separate the real and imaginary parts of the input $\mathbb{X}_{en}$ and get through the linear transform Linear to form the relative $\mathbb{Q}$, $\mathbb{K}$, $\mathbb{V}$. For the simplicity, we only demonstrate the $\mathbb{Q}$ below:

$$\mathbb{Q} = \text{Complex}(\text{Linear}(Re(\mathbb{X}_{en})), \text{Linear}(Im(\mathbb{X}_{en})). \tag{4.9}$$

With the Complex input $\mathbb{Q}$, $\mathbb{K}$, $\mathbb{V}$, we straightforwardly refine the Attention function to FHMA as presented:

$$\begin{aligned}\text{FMHA}(\mathbb{Q},\mathbb{K},\mathbb{V}) = \text{Complex}(&\text{Softmax}(Re(\frac{\mathbb{Q}\mathbb{K}^T}{\sqrt{d_{\mathbb{K}}}})),\\ &\text{Softmax}(Im(\frac{\mathbb{Q}\mathbb{K}^T}{\sqrt{d_{\mathbb{K}}}})) \cdot \mathbb{V}.\end{aligned} \tag{4.10}$$

Encoder/Decoder. As illustrated in Fig. 4.2, the encoder is assembled of a stack of N layers. Each layer in the encoder has two components: FMHA and FGCN. The FMHA in the encoder generates the attention units from the $\mathbb{X}_{en}$, and FGCN is to learn the representation in the embedding space further. AddNorm indicates LayerNorm(x + Sublayer(x)) which is a residual link design. The equations are specified as:

$$\begin{aligned}\mathbb{X}_{en}^{l,1}, \_ &= \text{AddNorm}(\text{FMHA}(\mathbb{X}_{en}^{l-1,1}) + \mathbb{X}_{en}^{l-1,1})\\ \mathbb{X}_{en}^{l,2}, \_ &= \text{AddNorm}(\text{FGCN}(\mathbb{X}_{en}^{l,1}) + \mathbb{X}_{en}^{l,1}),\end{aligned} \tag{4.11}$$

where "_" is denoted as the void attention output. $\mathbb{X}_{en}^{l} = \mathbb{X}_{en}^{l}, l \in \{1, \ldots, N\}$ stands for the output of the l-th encoder layer and $\mathbb{X}_{en}^{l,i}$ refers to the ith unit in the $\mathbb{X}_{en}^{l}$.

The decoder is constructed with a series of M stacked layers, with three elements in each layer: FMHA, FMHA and FGCN. The first FMHA unit is extract the latent information from the $\mathbb{X}_{de}$, while the second FMHA unit is to inference as:

$$\begin{aligned}\mathbb{X}_{de}^{l,1} &= \text{AddNorm}(\text{FMHA}(\mathbb{X}_{de}^{l-1,1}) + \mathbb{X}_{de}^{l-1,1})\\ \mathbb{X}_{de}^{l,2} &= \text{AddNorm}(\text{FMHA}(\mathbb{X}_{de}^{l,1}, \mathbb{X}_{en}^{N}) + \mathbb{X}_{de}^{l,1})\\ \mathbb{X}_{de}^{l,3} &= \text{AddNorm}(\text{FGCN}(\mathbb{X}_{de}^{l,2}) + \mathbb{X}_{de}^{l,2}),\end{aligned} \tag{4.12}$$

where "_" is denoted as the void attention output. $\mathbb{X}_{de}^{l} = \mathbb{X}_{de}^{l}, l \in \{1, \ldots, M\}$ represents the output of the l-th decoder layer and $\mathbb{X}_{de}^{l,i}$ refers to the ith unit in the $\mathbb{X}_{de}^{l}$.

4.6 Experiments

This section details the evaluation process for our Fre-Transformer comprehensive analysis. The in-depth analysis comprises benchmarks, experiment setup, performance evaluation, and sensitivity analysis, each detailed in the following subsections.

4.6.1 Dataset and Baselines

To evaluate the effectiveness of our model, we selected five financial time series forecasting models as baselines, including the vanilla deep sequential model and current state-of-the-art deep financial baselines, which are specifically:

- LSTM[35], where is a type of advanced, recurrent neural network (RNN) architecture used in the field of deep learning, capable of learning long-term dependencies in data sequences, incredibly efficient in financial deep learning.

- ALSTM[27], is an enhanced version of the traditional LSTM network, which incorporates an attention mechanism to improve the learning of long-term dependencies for more accurate stock market predictions.

- RSR[28] is a novel deep model for stock prediction named Relational Stock Ranking. It utilises the graph relation embedding within the LSTM framework to capture the cross-asset representation in Fin-MTS.

- HIST[78], where a graph-based framework forecasts stock trends by leveraging shared information across different stocks. The graph is organized around concept-oriented structures to enhance prediction performance.

- ESTIMATE[39], the newest deep learning model specially developed for stock movement prediction using attention mechanism onto an LSTM network with hypergraph and wavelet transform. It is abbreviated as ESTI in table 4.2,4.3.

To assess the performance of the proposed model, we utilise open-source real-world data from popular data provider Yahoo Finance. Ran Aroussi developed a threaded and Pythonic Library named yfinance, which allows us to download the specified open-source market data from Yahoo Finance[1]. For fairness and trustworthiness, we apply the most influential index, S&P500, which stands for the Standard and Poor's 500, ticker symbol ĜSPC. The dataset consists of three components: training dataset, validation dataset,

[1] http://finance.yahoo.com

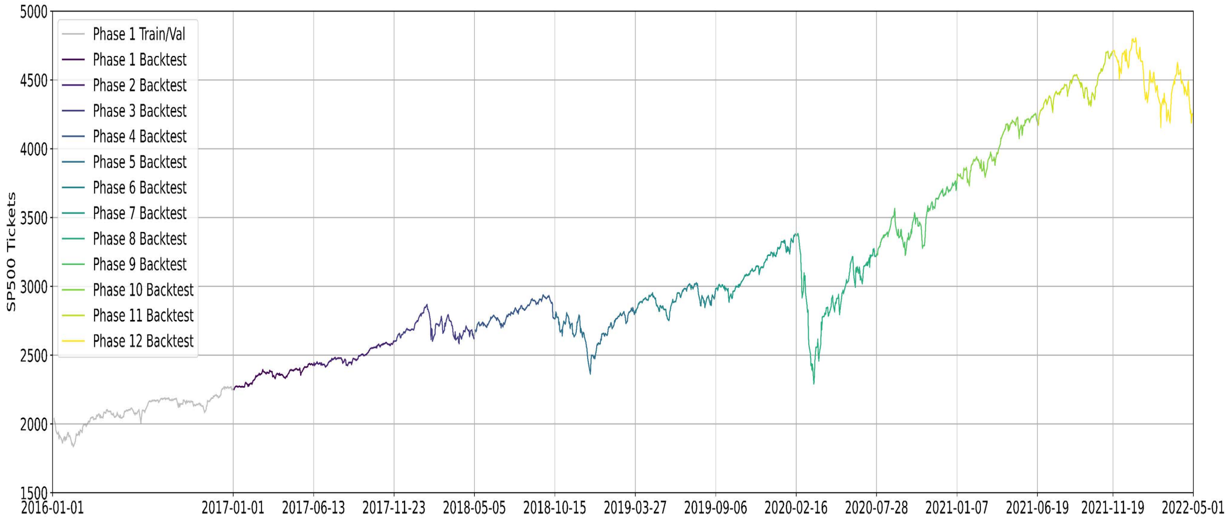


Figure 4.3: Arranged S&P500 dataset phases for experiments. The line indicates a real-world S&P 500 Index closing price. The grey line segment on the left side denotes the training and validation data in phase 1. The spacing between two adjacent grid lines on the x-axis corresponds to one phase period.

and backtest dataset. The entire dataset covers 2016/01/01 to 2022/06/01 (1593 trading days) and split the data into 12 phases[39, 78] due to the markets period with different representations, volatility, trading volume, and markets sentiment in Fig 4.3. Each phase contains training data with a duration of 10 months, a validation dataset with a duration of 2 months, and a backtest dataset with a duration of 6 months. For each day, S&P500 has six features, which are Open, High, Low, Close, Adj Close, and Volume.

4.6.2 Performance Metrics and Reproducibility Environment

Mainstream time series forecasting metrics, mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared error (MSE), and rooted mean squared error (RMSE), are not significant in financial data forecasting. We adopt the typical financial evaluation metrics, which are both quantitatively and qualitatively significant in finance, to gauge the accuracy of the experiment as details:

- *Information Coefficient (IC):* reflects the relationship of predictions to the ground truth, determined through the average Pearson correlation coefficient. IC = corr(R_p, R_a).

- *RankInformation Coefficient (Rank_IC):* is a variation of IC, calculated by the Spearman coefficient, evaluating the ranking of assets based on intra-series potential profit. Rank_IC = corr(rank(R_p), rank(R_a)).

- *Information ratio based IC (ICIR):* is a metric that combines two key aspects, accuracy and consistency of a financial forecaster, computed by dividing the average IC by the standard deviation of the IC. ICIR = mean(IC)/std(IC).

- *Rank information ratio based IC (Rank_ICIR):* is the information ratio of computed by average Rank_IC and standard deviation of the rank_IC[55]. Rank_ICIR = mean(Rank_IC)/std(Rank_IC).

Table 4.1: Datasets phases Arrangement.

Phases	Train	Val	Test
1	2016/1/1-2016/10/31	2016/11/1-2016/12/31	2017/1/1-2017/6/13
2	2016/6/12-2017/4/12	2017/4/13-2017/6/12	2017/6/13-2017/11/23
3	2016/11/22-2017/9/22	2017/9/23-2017/11/22	2017/11/23-2018/5/5
4	2017/5/4-2018/3/4	2018/3/5-2018/5/4	2018/5/5-2018/10/15
5	2017/10/14-2018/8/14	2018/8/15-2018/10/14	2018/10/15-2019/3/27
6	2018/3/26-2019/1/24	2019/1/25-2019/3/26	2019/3/27-2019/9/6
7	2018/9/5-2019/7/6	2019/7/7-2019/9/5	2019/9/6-2020/2/16
8	2019/2/15-2019/12/16	2019/12/17-2020/2/15	2020/2/16-2020/7/28
9	2019/7/28-2020/5/27	2020/5/28-2020/7/27	2020/7/28-2021/1/7
10	2020/1/7-2020/11/6	2021/11/7-2021/1/6	2021/1/7-2021/6/19
11	2020/6/18-2021/4/18	2021/4/19-2021/6/18	2021/6/19-2021/11/19
12	2020/11/18-2021/9/18	2021/9/19-2021/11/18	2021/11/19-2022/5/1

Our experiments are standardized to ensure a certain level of reproducibility by a defined playing field with certain Python package versions. The versions selected for all implementations were Python 3.8.13, PyTorch 1.13.1, Numpy 1.22.3, CUDA toolkit 11.6.1, and scikit-learn 1.2.2. All experiments were conducted on an Intel(R) Xeon(R) Gold 6238R CPU @ 2.20GHz system with 180 GB of main memory and Quadro RTX 6000 graphic cards with driver version 525.105.17 and CUDA version 12.0. Furthermore, we integrated a module to control randomness, utilizing seed values to manage the stochasticity in various computational units, including the GPU, Python, and PyTorch.

4.6.3 Performance

The financial time series forecasting performance on the S&P500 dataset is detailed in Table 4.2, 4.3, with baselines comparative data originating from [39]. All results reported in Tables 4.2 and 4.3 were obtained with mean-squared error (MSE) as the training loss. In terms of `IC` and `Rank_IC` Metrics, our FreTransformer outperformed the current deep financial models[71] and achieved overall best performance across all the phases. The metrics `ICIR` and `Rank_ICIR` are utilised to respectively evaluate the level of randomness and trustworthiness in the predictive models for `IC` and `Rank_IC`. Our model realized overall best compared to the baseline models, featuring the greatest number of optimal `IC` and `Rank_IC` in conjunction with leading `ICIR` and `Rank_ICIR`. Specifically, with extraordinary stationary performance, our method significantly outperforms in the `ICIR` and `Rank_ICIR`, proving our contributions. On the other hand, with stable training and validation data, applying our model to a highly unstable backtest dataset will cause inferior performance. Our method fails to achieve the best `IC` and `Rank_IC` in some phases due to the training and validation periods being both highly volatile. This volatility is primarily attributed to the occurrence of black swan events, which significantly impacted market periodic dynamics. Our model limitation is neglecting the real-world black swan events, resulting in insufficient performance on the highly volatile phases. Despite that, it is worth remarking that naïve deep learning methods are still competitive in financial datasets in some phases. LSTM achieved the best `Rank_IC` performance in phases 8 and 11. ALSTM attained the peak `Rank_IC` achievement in phase 9. ESTIMATE is the top-performing model among the baselines, showing superior performance in phases 3 and 9. The reason is wavelet hypergraph attention in ESTIMATE captures both intra- and inter-series correlations. Overall, our model outperforms the baseline according to the mean values of the metrics:`IC` and `Rank_IC`.

4.6.4 Ablation Study

To address the Research Question: What impact does each component of the model have? We assessed the significance of each component within our model by developing four distinct variants: `(FT-1)` This variant eliminates the FGCN operators in the encoder layers within FreTransformer. `(FT-2)` This variant eliminates the FGCN operators in the decoder layers within FreTransformer. `(FT-3)` This variant removes the FGCN operators

[2] For each phase, the three best-performing methods are denoted using distinct markings: **bold** for the top method, superscript asterisk* for the second-best, and underline for the third-best. While Diff refers to the metrics difference between the first model and the model in the related column.

Table 4.2: Overall accuracy^2, IC.

Ms/Ps	**LSTM**	**ALSTM**	**RSR**	**HIST**	**ESTI**	**Ours**
1	0.014	-0.024	0.008	0.003	0.061*	**0.100**
2	-0.030	-0.025	-0.009	0.000	0.010*	**0.113**
3	-0.016	0.025*	-0.003	0.005	**0.134**	0.125*
4	0.006*	-0.009	-0.017	-0.010	-0.030	**0.124**
5	0.020	0.029*	-0.009	0.006	0.012	**0.112**
6	-0.034	-0.018	0.018*	0.008	0.003	**0.132**
7	-0.006	-0.033	0.011*	0.005	0.006	**0.094**
8	0.014*	-0.024	-0.005	-0.017	0.012	**0.074**
9	-0.002	0.045	-0.036	0.006	**0.160**	0.111*
10	-0.039	-0.046	0.018	0.009	0.031*	**0.096**
11	0.022	0.016	-0.058	0.011	0.043*	**0.126**
12	-0.023	-0.015	0.003	0.006	0.093*	**0.145**
IC$_{\text{std}}$	0.022	0.028	0.022	**0.008**	0.057	0.018*
IC$_{\text{mean}}$	-0.006	-0.007	0.003	0.003	0.045*	**0.113**
Diff$_{\text{IC}_\text{m}}$	0.000	-0.001	+0.009	+0.009	+0.051*	**+0.119**
ICIR	-0.282	-0.232	0.139	0.326	0.777*	**6.290**
Diff$_{\text{ICIR}}$	0.000	+0.050	+0.421	+0.608	+1.059*	**+6.572**

Table 4.3: Overall accuracy^2, Rank_IC.

Ms/Ps	**LSTM**	**ALSTM**	**RSR**	**HIST**	**ESTI**	**Ours**
1	-0.151	-0.211	0.031	0.085	0.040*	**0.100**
2	-0.356	-0.266	-0.018	-0.008	0.016*	**0.083**
3	-0.289	0.049	-0.005	0.125*	0.108	**0.129**
4	0.089*	-0.099	-0.033	-0.225	-0.019	**0.095**
5	0.186*	0.182	-0.009	**0.192**	0.026	0.116
6	-0.091	-0.289	0.029	**0.204**	0.016	0.146*
7	-0.151	-0.476	0.001	0.107	-0.014*	**0.194**
8	**0.201**	-0.243	-0.007	-0.328	-0.006	0.150*
9	-0.019	**0.242**	-0.019	0.174*	0.160	0.130
10	-0.496	-0.323	0.017	**0.256**	-0.002	0.170*
11	**0.259**	0.094	-0.072	0.215*	0.047	0.129
12	-0.397	-0.174	-0.031	0.157*	0.052	**0.196**
R_IC$_{\text{std}}$	0.251	0.267	0.228	0.181	0.053*	**0.034**
R_IC$_{\text{mean}}$	-0.101	-0.096	-0.007	0.080*	0.035	**0.137**
Diff$_{\text{R\_IC}_\text{m}}$	0.000	+0.005	+0.094	+0.181*	+0.136	**+0.238**
R_ICIR	-0.403	-0.360	-0.030	0.438	0.670*	**4.055**
Diff$_{\text{R\_ICIR}}$	0.000	+0.043	+0.373	+0.841	+1.073 *	**+4.458**

in the encoder embedding layers. (FT-4) This variant removes the FGCN operators in the decoder embedding layers. Table 5.3 demonstrates the detailed experiment results.

Table 4.4: Overall accuracy, IC, Rank_IC.

	FreTransformer	**FT-1**	**FT-2**	**FT-3**	**FT-4**
IC	**0.1449**	0.0144	0.0372	0.0887	0.1309
Rank_IC	**0.1956**	-0.0225	-0.0022	0.0734	0.1265

4.6.5 Sensitivity Analysis

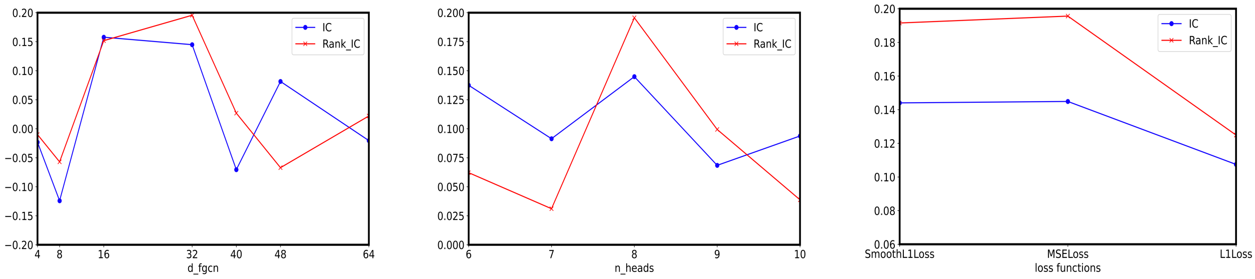
To meticulously evaluate the influence of hyperparameters within our proposed method, we carried out a thorough sensitivity study. This extensive analysis assesses a multitude of hyperparameter configurations across the entire end-to-end training process in phase 12.

- Primarily, we explore the FreTransformer unique hyperparameters crucial for the model performance, such as FGCN Dimensions d_fgcn in FGCN, Multi-head Attention n_heads, and reconstruction loss function.
- Furthermore, we analyse the regular hyperparameters in time series deep learning research questions, including the number of epochs, batch sizes, and learning rate.

In each experimental sequence, we rigorously test a set of designed hyperparameters across an available range, finding the best setting and showing the ascending and descending trends for each hyperparameter. We preserve the default setting for all of the other hyperparameters to distinctly evaluate the influence of each parameter on our model efficacy.

4.6.5.1 Effects of FreTransformer

Generally, the Transformer architecture has advantages in capturing complex and long data relationships by its advanced attention mechanisms embedded with deep neural networks to achieve great performance in sequential data. Nevertheless, the real-world financial data performance of Transformers in tasks such as time-series forecasting or anomaly detection can vary significantly due to the diversity of data distribution, task requirements, and the drift of the concept or data. To independently evaluate and



(a) FGCN Dimensions

(b) Multi-head Attention

(c) Loss Functions

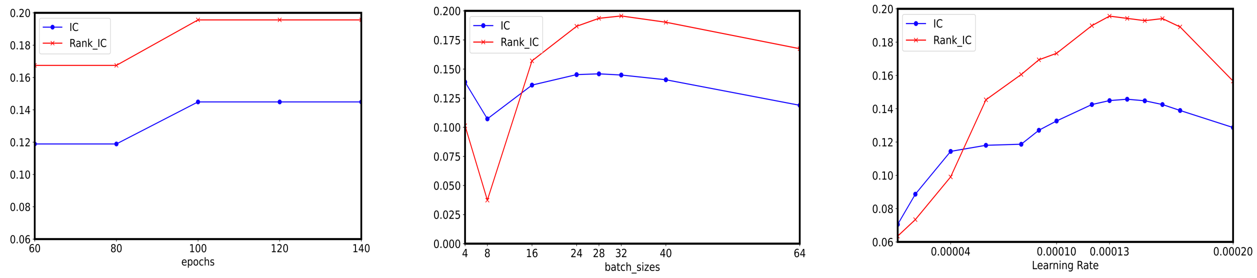
Figure 4.4: Sensitivity analysis of FreTrans related hyperparameters indicates significant findings: (a) shows that the FGCN operators d_ff in FreTrans profoundly influences outcomes, with optimal performance when the dimension ranges between 16 and 32. (b) demonstrates that n_heads exerts a notable effect on optimization, peaking in efficacy at 8. (c) indicates the most superior Reconstruction Loss Function is the SmoothL1Loss.

understand the impact of these elements on the performance of financial time series forecasting, we initiate a comprehensive sensitivity analysis, focusing on three essential hyperparameters.

FGCN Latent Dimensions. The dimensions of the FGCN Operators specify the density of latent information in FGCN that influences the model's capacity for capturing data representation in the embedding and encoder/decoder layer. The d_fgcn also determines the ability to seize the few shot financial daily data in the encoder/decoder layers. This hyperparameter is similar to the dimension of the feedforward network d_ff in the vanilla Transformer, which is vital in processing sequential data. To create an efficient FGCN Operator, we performed experiments that varied the dimensions of the latent dimension d_fgcn, specifically inspecting in {4,8,16,32,40,48,64}. These experiments' results and thorough analysis are presented in Fig. 4.4 (a), which highlights the optimal dimension of d_fgcn.

Multi-head Attention. The number of heads in the multi-head attention mechanism of FreTransformer, denoted as n_heads, is important in deciding the model's proficiency in interpreting diverse aspects of simultaneous input data. We chose to experiment with values in the set {6,7,8,9,10}, aiming to understand how varying n_heads affects the model's learning ability. The detailed outcomes are demonstrated in Fig. 4.4 (b).

Loss Functions. The choice of loss function in FreTransformer is a critical factor in defining the model's ability to learn from the training data accurately. We experimented with a range of loss functions, including Mean Squared Error (MSELoss), Mean Absolute Error (L1loss), and Huber Loss (SmoothL1loss). The comparison and analysis of three types of loss functions are visualised in Fig. 4.4 (c).



(a) Epochs

(b) Batch Sizes

(c) Learning Rate

Figure 4.5: Sensitivity analysis of time series deep learning related hyperparameters reveals critical outcomes: (a) The best performance was observed at 60 epochs. (b) A batch size of 4 yields optimal results. (c) The superior performance of the learning rate is revealed at 8×10^{-5}.

4.6.5.2 Effects of Deep Learning

In deep neural networks, the number of epochs, batch sizes, and learning rate count towards optimal performance.

Number of Epochs. The number of epochs in the FreTransformer impacts the performance in the training phase, so we test the model's forecasting capacity at epochs {60,80,100,120,140} to explore the consistency and randomness, while Fig. 4.5 (a) details the results and insights.

Batch Sizes. The batch size in FreTransformer significantly affects the model's learning dynamics and computational efficiency. Therefore, we tested various batch sizes {4,8,16,32,40,64} to find the gap between adequate and inadequate learning per experiment. Fig. 4.5 (b) illustrates the test findings and explorations.

Learning Rates. The learning rate in FreTransformer defines how quickly the gradient descended in the model. We experimented with a range of learning rates between 1×10^{-6} and 2×10^{-5} to find the stability of gradient updates and optimal experiment results. Fig.4.5 (c) shows the study results and reviews.

4.7 Summary

This chapter introduced FreTransformer, a Fourier-enhanced Transformer framework designed for financial multivariate time-series (Fin-MTS) forecasting. We first formalised the Fin-MTS prediction problem and motivated the need for frequency-domain modelling. The model architecture was then detailed, highlighting the Fourier Graph Convolution

Network (FGCN) for joint intra-/inter-series representation and the Fourier Multi-Head Attention mechanism that preserves long-range dependencies.

A comprehensive sensitivity analysis demonstrated how key hyperparameters,ÄîFGCN latent dimension, number of attention heads and learning-rate warm-up steps,Äîaffect predictive accuracy and computational cost. Extensive experiments on the S&P 500 dataset showed that FreTransformer achieves state-of-the-art performance, surpassing established baselines in IC, Rank IC, ICIR and Rank ICIR metrics while maintaining robustness across multiple market phases. We also analysed failure cases in highly volatile "black-swan" periods, outlining how market regime shifts challenge fully connected graphs and suggesting sparsification masks as a mitigation strategy.

Where FreTransformer focuses on numerical Fin-MTS alone, Chapter 5 extends the framework cross-modally, showing how news sentiment and analyst reports can further enhance forecasts when combined with the frequency-domain backbone.

Overall, FreTransformer provides a scalable, interpretable and empirically strong solution for Fin-MTS forecasting, laying the groundwork for the cross-modal LLM extensions explored in the following chapter.

CHAPTER 5

Fine-Tuned Large Language Model for Financial Time Series Forecasting with FTS-Text Embeddings

Financial time series forecasting is significantly importance in many real-world tasks and has been extensively studied. However, current forecasting methods often lack the ability to effectively integrate both time series data and relevant textual information, limiting their ability to fully capture complex financial patterns and market sentiment. Additionally, existing approaches struggle with accurately predicting rare, high-impact events like black swan occurrences. In this chapter, we propose LLM4FT, a novel framework that leverages large language models (LLMs) to seamlessly integrate financial time series and textual data for enhanced forecasting accuracy. At the core of LLM4FT is the FTS-Text Embedder, which unifies time series data with its corresponding textual explanations, allowing the model to capture both intra-series patterns and contextual market insights. Furthermore, we introduce FinLoss, a specialised loss function designed to optimise the Information Coefficient (IC), enabling the model to handle complex financial events and improve overall prediction performance. Through comprehensive evaluations, we demonstrate that LLM4FT significantly outperforms state-of-the-art models, excelling in both general forecasting tasks and challenging scenarios involving rare market events. Our comprehensive evaluations demonstrate the potential of LLM4FT as a powerful tool for financial time series forecasting, offering robust performance across various financial environments.

5.1 Introduction

Forecasting financial time series with high accuracy is crucial for effective decision-making in areas such as portfolio management, risk mitigation, and economic prediction [21, 62, 65, 79]. However, the challenge of accurately forecasting financial time series stems from the inherent complexity of financial markets, which are often driven by a combination of historical data and external factors such as market sentiment, financial news, and rare events like black swan occurrences [73]. Traditional forecasting methods tend to fall short in capturing the intricate relationships between time series data and these external factors. These methods are often highly specialised and require significant domain expertise, making them less flexible and generalisable across different financial markets.

Recent advancements in large language models (LLMs) have shown that these models possess impressive capabilities in natural language processing (NLP), reasoning, and pattern recognition [59]. However, while LLMs have transformed tasks in NLP and other domains, their application in financial time series forecasting remains largely unexplored. Current approaches primarily focus on numerical data, missing out on the valuable insights embedded in accompanying textual data such as financial reports and news. Additionally, traditional models for time series prediction are not well-suited for integrating such multimodal data, which limits their overall performance, especially in volatile and unpredictable markets.

Addressing these challenges requires a model that can seamlessly integrate both time series data and its corresponding textual context to fully capture the dynamics of financial markets [10]. To this end, we propose a novel framework, LLM4FT (Large Language Model for Financial Time Series Forecasting), which leverages the capabilities of pre-trained LLMs as its backbone and introduces a specialised FTS-Text Embedder. This embedder processes time series data alongside textual explanations and unifies them into a single embedding. The FTS-Text Embedder first normalises the time series data and embeds the textual information. These embeddings are then combined using attention mechanisms, which allow the model to capture both intra-series dependencies and the contextual insights provided by the text. By fusing these two modalities, LLM4FT can effectively model complex patterns and relationships that would otherwise be missed by traditional models.

In addition to this, we introduce a custom FinLoss function, which is specifically designed to optimise the Information Coefficient (IC) during the fine-tuning phase of the model. The FinLoss function allows LLM4FT to better capture relationships between financial time series data and market-related events, such as black swan occurrences. By optimising IC, LLM4FT is capable of improving the accuracy of its predictions and handling volatile market conditions more effectively.

In summary, this chapter addresses the key challenges of forecasting financial time series data by introducing a novel framework that seamlessly integrates textual information with time series data through the FTS-Text Embedder and optimises forecasting performance using the specialised FinLoss function. The contributions of this chapter are as follows: (1) LLM4FT is the first framework that leverages a pre-trained LLM as its backbone for financial time series forecasting, enabling the model to learn from both numerical and textual data simultaneously. (2) The FTS-Text Embedder combines the normalised time series data with textual explanations, capturing both the intra-series patterns and the contextual meaning of the text. This modular embedder can be integrated into existing LLM architectures, such as LLaMA or GPT, to predict future financial values directly. (3) The FinLoss function is introduced to optimise the Information Coefficient (IC), enhancing the model's ability to handle rare market events and improve the overall prediction accuracy in volatile conditions.

The remainder of this chapter is structured as follows: Section 5.2 presents the LLM4FT framework in detail, beginning with an overview of its structure, followed by a deep dive into the FTS-Text Embedder and the custom FinLoss function. Section 5.3 details the experiments conducted to evaluate the model, demonstrating the effectiveness of LLM4FT in various financial forecasting tasks. Finally, Section 5.4 provides a summary of the key insights and findings from this chapter, highlighting the potential for future research and improvements in leveraging LLMs for financial time series forecasting.

5.2 Problem definition

Fin-MTS with end-to-end text embedding. Let

$$\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{L\times N}, \qquad \mathcal{T}_{t-L+1:t} = \{\tau_{t-L+1}, \dots, \tau_t\}$$

denote an L-step look-back window of structured Fin-MTS features and the corresponding raw textual tokens (news/analyst notes).

A learnable text-embedding module g_{θ_e} maps the tokens to embeddings $\mathbf{E}_{t-L+1:t} = g_{\theta_e}(\mathcal{T}_{t-L+1:t}) \in \mathbb{R}^{L\times E}$, which are not pre-aligned but trained jointly with the main forecaster f_{θ_f}. The forecasting objective becomes

$$\hat{\mathbf{Y}}_{t+1:t+\tau} = f_{\theta_f}\Big(\mathbf{X}_{t-L+1:t},\, g_{\theta_e}(\mathcal{T}_{t-L+1:t})\Big) \in \mathbb{R}^{\tau \times N},$$

where $\theta = \{\theta_e, \theta_f\}$ is learned end-to-end.

5.3 Large Language Model for Financial Time Series Forecasting

This section outlines the architecture of the LLM4FT framework, which leverages a pre-trained large language model (LLM) as its backbone to process both financial time series data and corresponding textual information. The time series data is first normalised and transformed into a suitable format, while the textual data is tokenised and embedded. A key component of this process is the FTS-Text Embedder, described in Section 5.2.1, which ensures the proper alignment of the two data modalities for the model. Attention mechanisms are then employed to effectively capture intra-series dependencies within the time series data as well as the interactions between the time series and textual context. This enables the model to focus on the most critical patterns and relationships, enhancing its forecasting capabilities. Because newswire sentiment and analyst reports can be systematically skewed-bullish in rising markets, sparse on small-cap stocks, and subject to publication lags the joint-attention block applies a confidence-weighted mask that down-weights texts with low coverage consistency, thereby reducing bias propagation into the forecast layer. Finally, the framework integrates the processed data through a projection layer to generate predictions, while maintaining the integrity of the pre-trained LLM, allowing for flexible and accurate financial forecasting across different tasks.

5.3.1 FTS-Text Embedder

The FTS-Text Embedder processes and aligns the time series data with three types of embeddings: the time series embedding, the dataset description embedding, and the LLM text embedding. These embeddings are then combined to provide a unified representation suitable for input into the LLM.

Financial Time Series Embedding. Let $\mathbf{X} \in \mathbb{R}^{N \times T}$ represent the financial time series data, where T is the number of time steps and N is the dimensionality of the financial time series features. We can get the each financial time series is $\mathbf{X}^{(i)} \in \mathbb{R}^{1\times T}$. The time series data is divided into consecutive, possibly overlapping *input windows* to prevent the distribution drift[45], where each window has a length L_w. The total number of windows is given by:

$$N_w = \left\lfloor \frac{T - L_w}{S} \right\rfloor + n,$$

where S represents the sliding step between consecutive windows, and n is an adjustment factor accounting for boundary conditions[57]. This embedding creates a data augmentation method for LLM to learn the time series input more robustly.

Given these windows $\mathbf{X}_W^{(i)} \in \mathbb{R}^{N_w \times L_w}$, where i denotes the i-th time series, each window is embedded into the latent space representation $\hat{\mathbf{X}}_W^{(i)} \in \mathbb{R}^{N_w \times d_m}$ using a simple linear layer as the embedder:

$$\hat{\mathbf{X}}_w^{(i)} = \mathbf{X}_w^{(i)} \mathbf{W}_w + \mathbf{b}_w,$$

where $\mathbf{W}_w \in \mathbb{R}^{L_w \times d_w}$ is the learnable weight matrix and d_m is the output embedding dimension. This process is repeated for each i, producing embedded windows across all time series data, which are then used for further modelling tasks.

Dataset Description Embedding. Let $\mathbf{D} \in \mathbb{R}^{M_{description} \times d_{description}}$ denote the dataset description data, where M is the number of description tokens, and $d_{description}$ is the dimensionality of the description embedding. This description is processed similarly to the time series embedding:

$$\hat{\mathbf{D}} = \mathbf{D}\mathbf{W}_d + \mathbf{b}_m,$$

where $\mathbf{W}_d \in \mathbb{R}^{d_{description} \times d_m}$ is the learnable weight matrix, ensuring that the dataset description embedding dimension matches the time series embedding. The resulting embedding $\hat{\mathbf{D}} \in \mathbb{R}^{M_{description} \times d_m}$ shares the same dimensionality as the time series embedding, allowing for seamless integration in subsequent layers. This alignment enables the model to capture both numerical and descriptive aspects of the dataset effectively.

LLM Text Embedding. The textual data, which includes pre-trained LLM embeddings, is tokenised and embedded using a pre-trained LLM. Let $\mathbf{T} \in \mathbb{R}^{M_{LLM} \times d_{llm}}$ represent the tokenised text, where N is the number of tokens, and d_v is the dimensionality of the LLM embeddings. The LLM processes the tokens to produce:

$$\hat{\mathbf{T}} = \mathbf{T}\mathbf{W}_t + \mathbf{b}_m,$$

where $\mathbf{W}_t \in \mathbb{R}^{d_{llm} \times d_m}$ is the learnable weight matrix, ensuring that the dataset description embedding dimension matches the time series embedding. This produces textual embeddings $\hat{\mathbf{T}} \in \mathbb{R}^{M_{LLM} \times d_m}$ that are aligned with the dimensions of the other embeddings.

Combined Embedding. The combined embedding process integrates multiple sources of information, including the embedded time series patches $\hat{\mathbf{X}}_P$, dataset description embeddings $\hat{\mathbf{D}}$, and LLM-generated text embeddings $\hat{\mathbf{T}}$. These embeddings are concatenated and processed using a multi-head attention mechanism, enabling the model to capture dependencies across different modalities. The multi-head attention is applied as follows:

$$\mathbf{Z}^{(i)} = \text{MultiHeadAttention}(\mathbf{Q}^{(i)}, \mathbf{K}, \mathbf{V}) = \text{MultiHeadAttention}(\hat{\mathbf{X}}_w^{(i)}, \hat{\mathbf{D}}, \hat{\mathbf{D}}),$$

where $\mathbf{Z}^{(i)} \in \mathbb{R}^{L_w \times d_m}$ represents the combined embedding for the i-th time series window, with L being the sequence length and d_m the embedding dimension. The query $\mathbf{Q}^{(i)} = \hat{\mathbf{X}}_w^{(i)}$ corresponds to the i-th time series patch, while the keys and values, $\mathbf{K} = \hat{\mathbf{D}}$ and $\mathbf{V} = \hat{\mathbf{D}}$, are derived from the dataset description embeddings. For each attention head, the attention is computed as:

$$\mathbf{Z}_{(j)}^{(i)} = \text{Attention}(\mathbf{Q}_{(j)}^{(i)}, \mathbf{K}_{(j)}, \mathbf{V}_{(j)}),$$

where $\mathbf{Z}_{(j)}^{(i)} \in \mathbb{R}^{d_m}$ is the output for the j-th head. The final combined embedding $\mathbf{Z} \in \mathbb{R}^{L \times d_m}$ is the result of concatenating the outputs from all attention heads. This embedding captures both temporal dependencies from the time series and contextual information from the dataset description, improving the accuracy of financial time series predictions.

The final combined embedding is formed by concatenating the LLM-generated text embeddings $\hat{\mathbf{T}}$ and the time series embedding $\mathbf{Z}$, which incorporates both the temporal dependencies from the time series and the contextual information from the dataset description. This operation is represented as follows:

$$\mathbf{Z}^{\hat{(i)}} = [\hat{\mathbf{T}}, \mathbf{Z}^{(i)}],$$

where $\hat{\mathbf{T}} \in \mathbb{R}^{M_{LLM} \times d_m}$ represents the LLM Text embeddings, and $\mathbf{Z}^{(i)} \in \mathbb{R}^{L_w \times d_m}$ is the output of the multi-head attention mechanism for the time series data. The concatenation is performed along the sequence dimension, resulting in: $\hat{\mathbf{Z}}^{(i)} \in \mathbb{R}^{(L_w + M_{LLM}) \times d_m}$ This combined embedding $\hat{\mathbf{Z}}^{(i)}$ is then passed to the LLM for further processing.

5.3.2 FinLoss

FinLoss is a specialised loss function designed to optimise the Information Coefficient (IC) for financial time series predictions. The IC measures the correlation between predicted values and target values, which is crucial in financial forecasting to capture patterns that are predictive of future movements.

The IC is computed as:

$$\mathrm{IC} = \frac{\sum(y-\overline{y})(\hat{y}-\overline{\hat{y}})}{\sqrt{\sum(y-\overline{y})^2}\cdot\sqrt{\sum(\hat{y}-\overline{\hat{y}})^2}},$$

where y represents the predicted values and $\hat{y}$ represents the true target values. The IC reflects how well the model predictions are aligned with the targets. The FinLoss function is structured to maximise this correlation while penalising negative IC values.

The loss is defined as:

$$L = (\sqrt{\sum(y-\overline{y})^2}\cdot\sqrt{\sum(\hat{y}-\overline{\hat{y}})^2}) - \sum(y-\overline{y})(\hat{y}-\overline{\hat{y}}) + \lambda\cdot\mathrm{ReLU}(-\sum(y-\overline{y})(\hat{y}-\overline{\hat{y}})),$$

where the regularisation term λ penalises negative IC values, ensuring a positive correlation between predictions and targets.

Rationale for Subtraction: The choice to use subtraction (denominator − numerator) rather than division (numerator/denominator) in the loss function is motivated by several factors. First, this approach enhances numerical stability. Division can lead to instability when the denominator approaches very small values or zero, which is a common occurrence in financial time series data due to fluctuations or low variability. Subtraction mitigates the risk of extreme loss values or undefined behaviour under such conditions.

Furthermore, subtraction leads to smoother gradient flow during backpropagation. When using division, the gradient becomes highly sensitive if the denominator is small, potentially resulting in exploding gradients. By using subtraction, the gradient remains more stable and predictable, improving convergence during optimisation processes such as stochastic gradient descent or Adam.

Additionally, subtraction directly penalises negative correlations between predictions and target values. When predictions y are negatively correlated with the true targets $\hat{y}$, the numerator becomes negative, and subtracting this value increases the loss, thereby penalising the model for incorrect predictions. This penalty is more effectively captured through subtraction than a division-based approach, where the negative correlation would not result in as sharp an increase in the loss value.

Jacobian Calculation: To compute the gradients of this loss with respect to the model's parameters, we compute the Jacobian matrix, $\mathbf{J}_{\text{Loss}}$, which captures the partial derivatives of the loss with respect to each model parameter:

$$\mathbf{J}_{\text{Loss}} = \frac{\partial \text{Loss}}{\partial \theta},$$

where θ represents the set of all model parameters (e.g., weights and biases). The Jacobian matrix provides the necessary information to update the model parameters during backpropagation, enabling gradient-based optimisation methods to minimise the FinLoss.

Moreover, the inclusion of regularisation terms, L_1 and L_2, ensures that the model parameters remain regularised. These terms are controlled by the hyperparameters λ_1 and λ_2, respectively, which modify the gradient contributions of the parameters:

$$L_{fin} = L + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta\|_2^2.$$

Thus, FinLoss not only optimises the model for financial time series predictions by maximising IC but also computes the gradients and regularises the model to avoid overfitting.

5.3.3 LLM4FT

The LLM4FT (Large Language Model for Financial Time Series Forecasting) architecture is designed to integrate diverse data sources, including financial time series, dataset descriptions, and contextual information from language models (LLMs), into a unified framework for improved prediction accuracy. As illustrated in Figure 5.1, the LLM4FT framework comprises several key components:

Input Data. LLM4FT processes three types of inputs: (1) time series data (e.g., S&P500), (2) data descriptions, which provide metadata about the financial time series, and (3) LLM Text embeddings, derived from contextual textual information. These data streams allow the model to leverage both numerical and textual insights, enhancing its ability to capture market trends and anomalies.

FTS-Text Embedder. The FTS-Text Embedder aligns and processes the inputs by creating three types of embeddings: financial time series embeddings, dataset description embeddings, and LLM Text embeddings. The time series data is first normalised and segmented into sliding windows, enabling the model to capture local temporal patterns and reduce distribution shift issues. Simultaneously, LLM Text is tokenised and embedded, producing text representations that are compatible with time series embeddings.

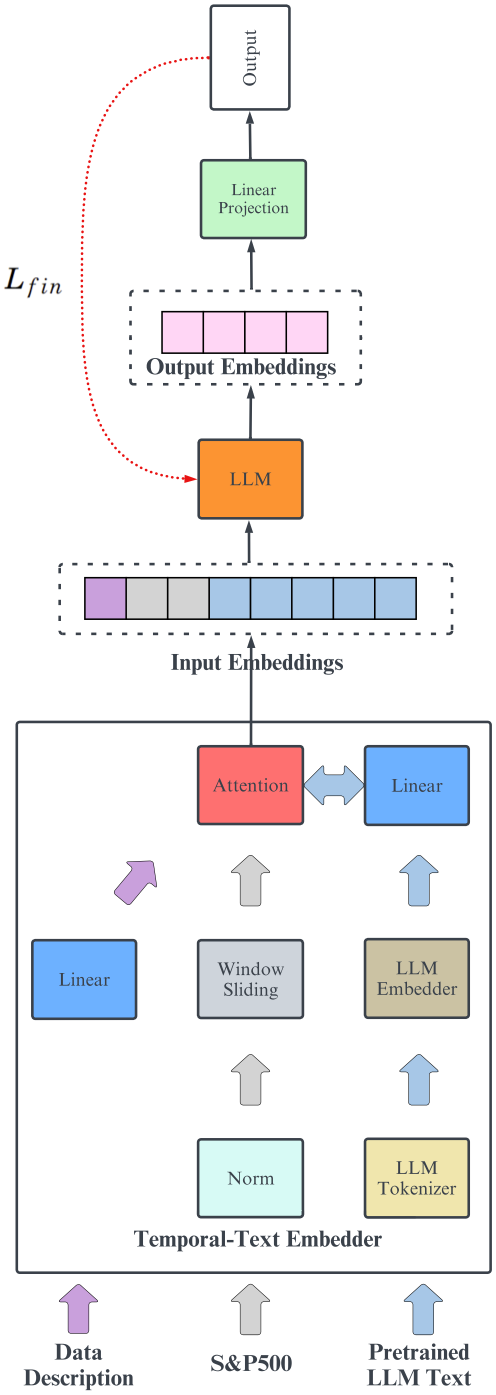


Figure 5.1: Architecture of LLM4FT. The model processes three inputs: a data description, financial time series data (e.g., S&P500), and LLM Text for contextual information. In the Temporal-Text Embedder, the time series data is normalised, segmented into sliding windows, and processed with linear and attention layers. Concurrently, LLM Text is tokenised and embedded, then combined with time series embeddings to form unified input embeddings. These are fed into the LLM, producing output embeddings, which are linearly projected to yield final predictions. The entire architecture is optimised with the FinLoss function, L_{fin}, enhancing financial forecasting accuracy.

The outputs from this module form unified input embeddings that combine both time series and textual data, preparing them for further processing.

Multi-Head Attention and LLM Processing. The unified embeddings are then fed into a multi-head attention mechanism to capture dependencies across the different modalities. This mechanism enables the model to focus on relevant patterns within the time series data, the dataset descriptions, and the LLM text embeddings. The combined embeddings are concatenated with the LLM-generated text embeddings and subsequently input into the LLM backbone. Here, the model further processes the data to produce output embeddings that synthesise the financial and textual information.

Output and FinLoss Optimisation. The output embeddings are then projected linearly to yield the final predictions. The model is optimised using the specialised **FinLoss** function, denoted as L_{fin}, which is tailored to financial forecasting by optimising the Information Coefficient (IC) and ensuring positive correlations between the predicted values and the actual target values. FinLoss also includes regularisation terms to prevent overfitting, thus enhancing the model's robustness.

Overall, LLM4FT leverages both the temporal patterns in financial time series data and the contextual insights from textual data, enabling accurate and reliable financial forecasting.

5.4 Experiments

This section details the evaluation process for our LLM4FT comprehensive analysis of three research questions, which exhibit as follows:
(RQ1) Can our proposed FreTransformer model outperform state-of-the-art Fin-MTS prediction solutions?
(RQ2) What impact does each component of the model have?
(RQ3) Does our model display hyperparameter sensitivity?
The in-depth analysis comprises benchmarks, experiment setup, performance evaluation, and sensitivity analysis, each detailed in the following subsections.

5.4.1 Benchmarks

To evaluate the effectiveness of our model, we selected five financial time series forecasting models as benchmarks, including the vanilla deep sequential model and current state-of-the-art deep financial baselines, which are specifically:

- LSTM [35], where is a type of advanced, recurrent neural network (RNN) architecture used in the field of deep learning, capable of learning long-term dependencies in data sequences, incredibly efficient in financial deep learning.

- ALSTM [27], is an enhanced version of the traditional LSTM network, which incorporates an attention mechanism to improve the learning of long-term dependencies for more accurate stock market predictions.

- RSR [28] is a novel deep model for stock prediction named Relational Stock Ranking. It utilises the graph relation embedding within the LSTM framework to capture the cross-asset representation in Fin-MTS.

- HIST [78], where a graph-based framework forecasts stock trends by leveraging shared information across different stocks. The graph is organised around concept-oriented structures to enhance prediction performance.

- ESTIMATE [39], the newest deep learning model specially developed for stock movement prediction using attention mechanism onto an LSTM network with hypergraph and wavelet transform. It is abbreviated as ESTI in table 4.2,4.3.

- FreTransformer [84], a novel deep learning model designed for financial time series forecasting by integrating Fourier Graph Convolution Network (FGCN) with a Transformer architecture. It effectively captures both temporal and frequency-domain patterns, making it particularly suitable for modelling complex financial time series. In tables 4.2 and 4.3, it is abbreviated as FreTrans.

5.4.2 Experiment Setup

Datasets. To assess the performance of the proposed model, we utilise open-source real-world data from popular data provider Yahoo Finance. Ran Aroussi developed a threaded and Pythonic Library named yfinance, which allows us to download the specified open-source market data from Yahoo Finance[1]. For fairness and trustworthiness, we apply the most influential index, S&P500, which stands for the Standard and Poor's 500, ticker symbol ĜSPC. The dataset consists of three components: training dataset, validation dataset, and backtest dataset. The entire dataset covers 2016/01/01 to 2022/06/01 (1593

[1] `http://finance.yahoo.com`

trading days) and split the data into 12 phases [39, 78] due to the markets period with different representations, volatility, trading volume, and markets sentiment in Fig 4.3. Each phase contains training data with a duration of 10 months, a validation dataset with a duration of 2 months, and a backtest dataset with a duration of 6 months. For each day, S&P500 has six features, which are Open, High, Low, Close, Adj Close, and Volume.

- **Source & Access:** Daily S&P 500 data (`ĜSPC`) fetched via the `yfinance`[2] Python wrapper.
- **Coverage:** 1,593 trading days spanning 2016-01-01 - 2022-06-01, split into 12 market phases to capture regime shifts (see Fig. 4.3; methodology follows [39, 78]).
- **Per-phase partitioning:** Training = 10 months, Validation = 2 months, Back-test = 6 months.
- **Daily feature vector ($F = 6$):** Open, High, Low, Close, Adjusted Close, Volume.

Performance Metrics. Mainstream time series forecasting metrics, mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared error (MSE), and rooted mean squared error (RMSE), are not significant in financial data forecasting. We adopt the typical financial evaluation metrics, which are both quantitatively and qualitatively significant in finance, to gauge the accuracy of the experiment as details:

- *Information Coefficient (IC):* reflects the relationship of predictions to the ground truth, determined through the average Pearson correlation coefficient.
- *RankInformation Coefficient (Rank_IC):* is a variation of IC, calculated by the Spearman coefficient, evaluating the ranking of assets based on intra-series potential profit.
- *Information ratio based IC (ICIR):* is a metric that combines two key aspects, accuracy and consistency of a financial forecaster, computed by dividing the average IC by the standard deviation of the IC.
- *Rank information ratio based IC (Rank_ICIR):* is the information ratio of computed by average Rank_IC and standard deviation of the rank_IC.

[2] `http://finance.yahoo.com`

Reproducibility environment. All experiments in this chapter use LLaMA-3-80B as the fine-tuned backbone and other open-source LLMs such as QianWen were not evaluated and remain future work. Our experiments are standardised to ensure a certain level of reproducibility by a defined playing field with certain Python package versions. The versions selected for all implementations were Python 3.11.9, PyTorch 2.2.2, Numpy 1.23.5, CUDA toolkit 12.1.105, and scikit-learn 1.2.2. All experiments were conducted on an AMD EPYC 9254 24-Core Processor CPU @ 2.90GHz system with 188 GB of main memory and two NVIDIA L40 48GB graphic cards with driver version 550.100 and CUDA version 12.4. Furthermore, we integrated a module to control randomness, utilising seed values to manage the stochasticity in various computational units, including the GPU, Python, and PyTorch. We also use DeepSpeed to optimise and scale large language models for faster training and lower resource consumption.

Table 5.1: Overall accuracy[2], IC.

Ps/Ms	**LSTM**	**ALSTM**	**RSR**	**HIST**	**ESTI**	**FreTrans**	**Ours**
1	0.014	-0.024	0.008	0.003	0.061	0.100*	**0.368**
2	-0.030	-0.025	-0.009	0.000	0.010	0.113*	**0.393**
3	-0.016	0.025	-0.003	0.005	**0.134**	0.125	0.132*
4	0.006	-0.009	-0.017	-0.010	-0.030	0.124*	**0.201**
5	0.020	0.029	-0.009	0.006	0.012	0.112*	**0.240**
6	-0.034	-0.018	0.018	0.008	0.003	0.132*	**0.172**
7	-0.006	-0.033	0.011	0.005	0.006	0.094*	**0.312**
8	0.014	-0.024	-0.005	-0.017	0.012	0.074*	**0.097**
9	-0.002	0.045	-0.036	0.006	0.160*	0.111	**0.196**
10	-0.039	-0.046	0.018	0.009	0.031	0.096*	**0.286**
11	0.022	0.016	-0.058	0.011	0.043	0.126*	**0.228**
12	-0.023	-0.015	0.003	0.006	0.093	0.145*	**0.151**
ICIR	-0.282	-0.232	0.139	0.326	0.777	**6.290**	2.608*

[2] For each phase, the three best-performing methods are denoted using distinct markings: **bold** for the top method, superscript asterisk* for the second-best, and underline for the third-best. While Diff refers to the metrics difference between the first model and the model in the related column.

Table 5.2: Overall accuracy[2], Rank_IC.

Ps/Ms	LSTM	ALSTM	RSR	HIST	ESTI	FreTrans	Ours
1	-0.151	-0.211	0.031	0.085	0.040	0.100*	**0.311**
2	-0.356	-0.266	-0.018	-0.008	0.016	0.083*	**0.348**
3	-0.289	0.049	-0.005	0.125	0.108	0.129*	**0.149**
4	0.089	-0.099	-0.033	-0.225	-0.019	0.189*	**0.199**
5	0.186	0.182	-0.009	0.192*	0.026	0.116	**0.208**
6	-0.091	-0.289	0.029	**0.204**	0.016	0.146	0.152*
7	-0.151	-0.476	0.001	0.107	-0.014	0.194*	**0.290**
8	**0.201**	-0.243	-0.007	-0.328	-0.006	0.150*	0.084
9	-0.019	**0.242**	-0.019	0.174*	0.160	0.130	0.162
10	-0.496	-0.323	0.017	0.256*	-0.002	0.170	**0.285**
11	**0.259**	0.094	-0.072	0.215*	0.047	0.129	0.201
12	-0.397	-0.174	-0.031	0.157*	0.052	**0.196**	0.110
R_ICIR	-0.403	-0.360	-0.030	0.438	0.670	**4.055**	2.605*

5.4.3 Performance Analysis

The financial time series forecasting performance on the S&P500 dataset is detailed in Table 4.2 and Table 4.3, with comparative baselines derived from [39]. In terms of the IC and Rank_IC metrics, our LLM4FT model outperformed existing deep financial models [71] and achieved the overall best performance across all phases. The metrics ICIR and Rank_ICIR are used to evaluate the degree of randomness and trustworthiness in the predictive models for IC and Rank_IC, respectively.

Our model demonstrated the overall best performance compared to the baselines, featuring the highest number of optimal IC and Rank_IC scores, along with leading ICIR and Rank_ICIR values. Specifically, with exceptional performance in stable conditions, our method significantly outperforms the baseline models in terms of ICIR and Rank_ICIR, highlighting the effectiveness of our contributions.

However, when applied to a highly unstable backtest dataset, our model exhibited inferior performance. It struggled to achieve the best IC and Rank_IC in certain phases due to both training and validation periods being highly volatile. This volatility is primarily attributed to the occurrence of black swan events, which disrupted market periodic dynamics. The model's limitation lies in its insufficient handling of real-world black swan events, leading to decreased performance during highly volatile phases.

Despite this, it is notable that simpler deep learning methods remain competitive in certain phases of the financial dataset. For instance, LSTM achieved the best `Rank_IC` performance in phases 8 and 11, while ALSTM attained the peak `Rank_IC` score in phase 9. Among the baselines, ESTIMATE is the top-performing model, showing superior performance in phases 3 and 9 due to its wavelet hypergraph attention mechanism, which effectively captures both intra- and inter-series correlations.

Impact of Pretrained LLM Embeddings. It is worth noting that the superior performance of LLM4FT in the initial phases can be partially attributed to the pretrained embeddings within the LLM component. These embeddings are likely to have captured a significant amount of relevant information about financial market dynamics during pretraining, such as common trends, volatility patterns, and event impacts, even though the pretraining was not explicitly focused on the S&P500 dataset. Consequently, the pre-trained embeddings provide a strong foundation for modelling, allowing LLM4FT to better understand and represent underlying relationships in the initial phases of time series forecasting.

The alignment between the pretrained LLM embeddings and the target financial data in the early phases facilitates effective representation learning, resulting in higher `IC` and `Rank_IC` scores. The embeddings act as a form of transfer learning, where knowledge gained from a broader corpus of financial text aids in improving predictive performance on the more structured financial time series data. This advantage, however, becomes less pronounced in the later phases, where the model encounters more challenging dynamics, including sudden shifts and black swan events, which may not be well captured by the pretrained embeddings alone.

Overall, our model outperforms the baselines according to the mean values of the metrics `IC` and `Rank_IC`, demonstrating the effectiveness of integrating LLM-based embeddings with financial time series forecasting.

5.4.4 Ablation Study

To address the Research Question (RQ2), we assessed the significance of each component within our model by developing four distinct variants: `(LLM4FT-1)` Remove the Temporal-Text Embedder entirely and use only the time series embeddings. This variant tests the importance of integrating textual information with financial time series data.

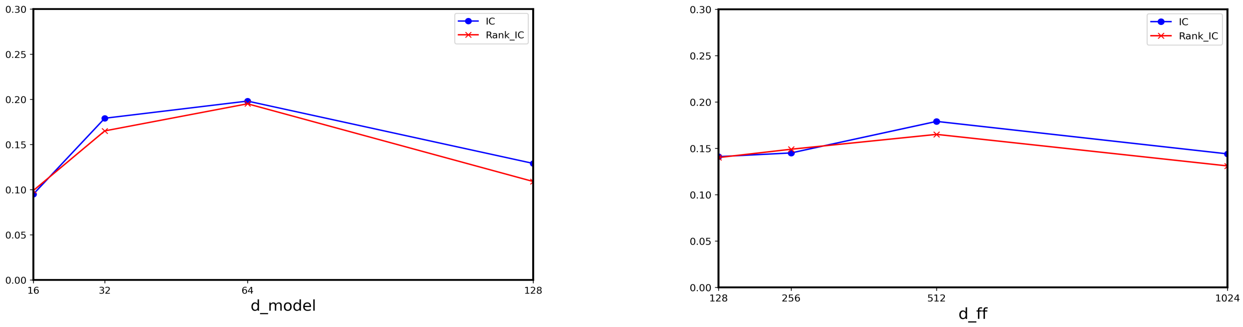
(LLM4FT-2) Include only the text embeddings with time series embeddings in the input. This evaluates how much predictive power is derived solely from the textual context. (LLM4FT-3) Use the time series embeddings alone with LLM embeddings to determine the standalone effectiveness of LLM embeddings input in the model. (LLM4FT-4) Replace FinLoss with a standard loss function like Mean Squared Error (MSE). This will help evaluate whether optimising Information Coefficient (IC) is superior to traditional loss functions in terms of predictive performance. Table 5.3 demonstrates the detailed experiment results.

Table 5.3: Overall accuracy, IC, Rank_IC.

	LLM4FT	**LLM4FT-1**	**LLM4FT-2**	**LLM4FT-3**	**LLM4FT-4**
IC	**0.368**	0.217	0.232	0.225	0.138
Rank_IC	**0.311**	0.184	0.215	0.208	0.101

5.4.5 Sensitivity Analysis

To comprehensively assess the impact of hyperparameters in the proposed LLM4FT model, we conducted an extensive sensitivity analysis, exploring a variety of hyperparameter configurations throughout the entire end-to-end training process.



(a) Feedforward Dimensions

(b) Model Dimensions

Figure 5.2: Sensitivity analysis of key hyperparameters in LLM4FT model: (a) Feedforward dimensions d_{model} show the highest performance at 64, with declining trends at lower and higher dimensions. (b) Model dimensions d_{ff} indicate optimal performance at 512, with lower and higher values resulting in reduced effectiveness.

- **Core LLM4FT Hyperparameters:** We focus on key hyperparameters unique to the LLM4FT architecture, including the dimensions of the model's attention layers

(d_{model}), the feedforward layers (d_{ff}), and the FinLoss-specific settings, such as the IC target and regularisation strength (λ).

- **FinLoss-Related Hyperparameters:** We explore how varying FinLoss parameters, such as IC target (IC$_{target}$) and the regularisation weight (λ), affect the model's ability to optimise for higher Information Coefficient (IC) and Rank IC.
- **General Training Hyperparameters:** We assess the impact of more conventional deep learning parameters, such as the number of training epochs, batch sises, and learning rates, to determine their influence on model convergence and predictive accuracy.

In each experimental run, we systematically test a range of hyperparameters while keeping others constant to isolate the influence of each parameter on the model's performance. This analysis provides insights into the optimal settings and illustrates how hyperparameters affect the predictive accuracy and robustness of LLM4FT.

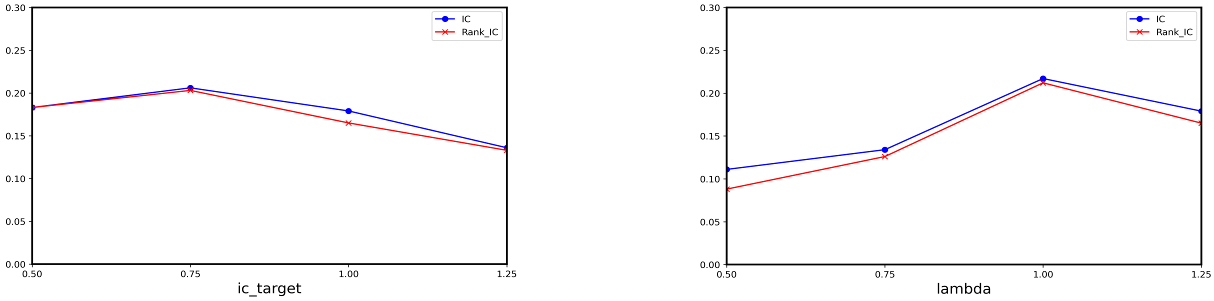
5.4.5.1 Effects of LLM4FT Components

LLM4FT introduces several novel components designed to enhance financial time series forecasting by integrating time series data with textual inputs. To evaluate their impact, we perform a detailed sensitivity analysis of three crucial hyperparameters.

Model Dimension in Temporal-Text Embedder. The model dimension d_{model} determines the sise of the vector representation used within the attention mechanism of the Temporal-Text Embedder. This dimension controls the capacity of the model to capture complex patterns in the combined time series and textual data. Higher values of d_{model} can improve the model's ability to learn intricate dependencies but also increase computational costs. We experimented with different values of d_{model}, specifically {16, 32, 64, 128}, finding the highest performance at 32. The results are presented in Fig. 5.2 (a).

Feedforward Layer Dimension in Temporal-Text Embedder. The feedforward layer dimension d_{ff} within the Temporal-Text Embedder determines the capacity of the model to process intermediate representations. It is typically set to a multiple of d_{model} and helps in transforming the attention outputs to more abstract representations. We tested different values of d_{ff}, such as {128, 256, 512, 1024}, and identified optimal performance at 512. The corresponding results are shown in Fig. 5.2 (b).

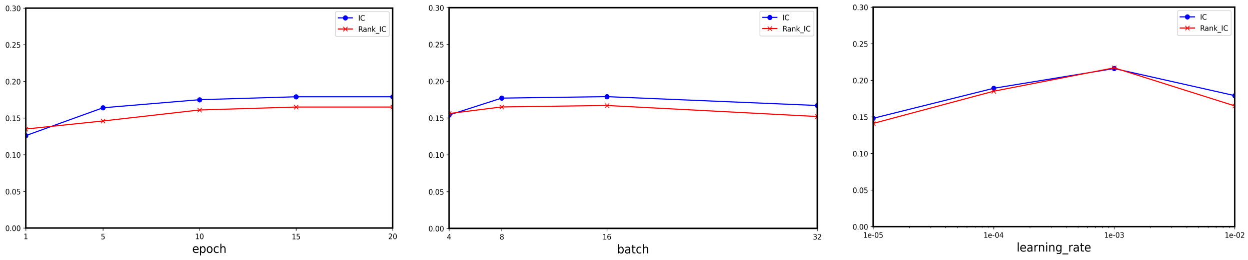
FinLoss Hyperparameters. The FinLoss function includes specific hyperparameters, such as the IC$_{target}$ and regularisation terms (λ). These control the focus on maximising the Information Coefficient (IC) while preventing overfitting. We tested variations in IC$_{target}$ {0.5, 0.75, 1.0} and regularisation strengths {0.5, 1.0, 1.25}, finding the optimal IC target at 0.75 and the best regularisation value at 1.0. The analysis results are presented in Fig. 5.3.



(a) IC Target

(b) Lambda

Figure 5.3: Sensitivity analysis of FinLoss hyperparameters: (a) The optimal IC target is around 0.75, yielding the highest IC and Rank_IC. (b) Increasing the value of λ to 1.25 results in a steady increase in both IC and Rank_IC, indicating the best regularisation strength.



(a) Epoch

(b) Batch Sizes

(c) Learning Rate

Figure 5.4: Sensitivity analysis of hyperparameters in time series deep learning for LLM4FT: (a) Optimal performance is achieved at 20 epochs, demonstrating the best IC and Rank IC results. (b) A batch sise of 8 results in peak IC and Rank IC performance, indicating the effectiveness of smaller batch sises. (c) The best learning rate is found to be 1×10^{-3}, yielding the highest IC and Rank IC values.

5.4.5.2 Effects of General Deep Learning Hyperparameters

As with other deep learning models, certain hyperparameters are essential to optimise the performance of LLM4FT.

Number of Epochs. We tested the model across a range of epochs {5, 10, 15, 20, 50}, with the best performance observed at 20 epochs. This indicates that the model achieves optimal convergence within this range, as presented in Fig. 5.4 (a).

Batch Sises. The batch sise influences both learning dynamics and computational efficiency. We experimented with various batch sises {4, 8, 16, 32}, identifying the batch sise of 4 as the most effective for maximising IC and Rank IC performance, as shown in Fig. 5.4 (b).

Learning Rates. We evaluated a range of learning rates, from 1×10^{-6} to 5×10^{-5}, with the optimal rate identified at 1×10^{-5}. This learning rate yielded the highest IC and Rank IC values, demonstrating stable gradient updates and convergence, as shown in Fig. 5.4 (c).

5.5 Summary

This chapter introduced LLM4FT, a novel framework that integrates large language models (LLMs) with financial time series data, addressing limitations in traditional forecasting models by capturing both numerical and contextual market insights. With the FTS-Text Embedder and the custom FinLoss function for optimising the Information Coefficient (IC), LLM4FT demonstrates superior predictive accuracy, especially in general market conditions, compared to existing models.

The experimental results and ablation study confirm the importance of multimodal data integration and highlight the pre trained LLM embeddings' role in improving model performance. However, challenges remain in highly volatile conditions. Future research could focus on refining the model's response to extreme events like black swan occurrences or incorporating additional data sources, such as social media sentiment, to enhance robustness across varying financial environments.

Part III

Part III

CHAPTER 6

Conclusions and Future Work

6.1 Conclusion

This thesis presents significant advancements in financial multivariate time series forecasting by introducing two novel frameworks: the Fourier Graph Convolution Transformer (FreTransformer) and a fine-tuned Large Language Model (LLM) with Financial Time Series-Text (FTS-Text) embeddings. The FreTransformer integrates graph knowledge embeddings and an innovative attention mechanism to capture complex dependencies within financial data effectively. Extensive experiments demonstrate its ability to model volatile, non-linear, and non-stationary financial time series, achieving superior performance compared to state-of-the-art models.

The fine-tuned LLM framework extends the application of LLMs to financial forecasting by incorporating textual and temporal data within a unified architecture. The proposed FTS-Text Embedder and FinLoss function enhance the model's capability to learn from diverse data sources, improving forecasting accuracy and interpretability. Collectively, these contributions address key challenges in financial time series forecasting, including dependency modelling, multi-modal data integration, and explainability, thereby setting a new benchmark in the domain.

6.2 Discussion and Future Work

6.2.1 Research Gaps Discussion

While the proposed methods provide significant advancements, some limitations remain. The FreTransformer primarily focuses on financial datasets with well-defined structures and may face challenges adapting to highly irregular or sparse data. Additionally, while the LLM framework leverages textual data effectively, the reliance on pre-trained language models may limit its performance when domain-specific data is scarce or insufficiently represented in the training corpus. Another notable gap is the computational complexity of both frameworks, which may hinder scalability to real-time applications or extremely large datasets.

6.2.2 Future Work

Future research can address these gaps by exploring several directions:

- **Scalability and Efficiency**: Optimising the computational efficiency of the FreTransformer and LLM frameworks to handle real-time financial forecasting tasks and large-scale datasets.
- **Cross-Market and Multi-Region Forecasting**: Extending the frameworks to analyse and forecast across multiple markets and regions, capturing interactions and dependencies at a global scale.
- **Adaptive Learning Mechanisms**: Incorporating adaptive learning techniques to dynamically update the models based on real-time data, enhancing their robustness to sudden market changes.
- **Enhanced Explainability**: Developing advanced explainability tools to provide deeper insights into model predictions, ensuring trust and transparency in decision-making processes.
- **Domain-Specific Pre-Training**: Fine-tuning LLMs with more extensive domain-specific datasets to improve their performance in underrepresented financial contexts, such as emerging markets or niche asset classes.

By addressing these research gaps and pursuing future directions, the methodologies proposed in this thesis can be further refined and extended, contributing to the evolving landscape of financial time series forecasting.

APPENDIX A

LIST OF ABBREVIATIONS

AE	AutoEncoder
ALSTM	Attention Long Short-Term Memory
ARIMA	AutoRegressive Integrated Moving Average
CAPM	Capital Asset Pricing Model
CCC	Constant Conditional Correlation
CNN	Convolutional Neural Network
CRRA	Constant Relative Risk Aversion
DCC	Dynamic Conditional Correlation
DNN	Deep Neural Network
EMD	Empirical Mode Decomposition
ESTIMATE	Efficient Stock Time-series Movement Integration with Attention-based Transformer and Hypergraph

FGCN	Fourier Graph Convolution Network
FFT	Fast Fourier Transform
Fin-MTS	Financial Multivariate Time Series
FMHA	Fourier Multi-Head Attention
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
GRU	Gated Recurrent Unit
IC	Information Coefficient
ICIR	Information Coefficient Information Ratio
iFFT	Inverse Fast Fourier Transform
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
RAG	Retrieval-Augmented Generation
Rank IC	Rank Information Coefficient
Rank ICIR	Rank Information Coefficient Information Ratio
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
S&P500	Standard & Poor's 500 Index
TFT	Temporal Fusion Transformer

TCN	Temporal Convolutional Network
VAE	Variational AutoEncoder
VMD	Variational Mode Decomposition

APPENDIX B

LIST OF NOTATIONS

X	Financial time series data, where T is the number of time steps and N is the dimensionality of the features.
X_t	Time-series input at time t, where $X_t \in \mathbb{R}^{T\times N}$.
Y_t	Forecasted values for the next τ time steps, $Y_t \in \mathbb{R}^{\tau\times N}$.
$\hat{Y}_t$	Predicted time-series output by the model.
f_θ	Forecasting function parameterized by θ.
FFT	Fast Fourier Transform, used to convert data from the time domain to the frequency domain.
iFFT	Inverse Fast Fourier Transform, used to revert frequency-domain data back to the time domain.
$A(X_t)$	Amplitude of the Fourier-transformed time series X_t.
$\Phi(X_t)$	Phase of the Fourier-transformed time series X_t.
G_W	Weighted graph $G_W = (X, A)$, where X is the node features and A is the adjacency matrix.
$\kappa[i,j]$	Green,Äôs kernel defined as $A_{ij} \circ W$, where W is the learnable weight matrix.

FGCN Fourier Graph Convolution Network, used to capture intra- and inter-series dependencies in the frequency domain.

Q, K, V Query, Key, and Value matrices used in the attention mechanism.

FMHA Fourier Multi-Head Attention, a variation of standard attention adapted for complex-valued inputs.

n_{heads} Number of attention heads in the multi-head attention mechanism.

d_{FGCN} Dimensionality of the latent representation learned by the FGCN.

σ Non-linear activation function applied to model outputs.

T, N T: Number of time steps; N: Number of features.

τ Forecasting horizon or the number of future time steps predicted.

ϵ Error term or noise in the forecasting process.

$X_W^{(i)}$ Windows of the time series data, where i is the index of the time series, W represents the window, N_w is the total number of windows, and L_w is the window length.

W_w Learnable weight matrix for the window embedding, transforming input time series data to latent space representation.

$Z^{(i)}$ Output of the multi-head attention mechanism for the time series data.

T LLM text embeddings that represent the textual data embeddings.

L_w Length of the sliding window used for time series data segmentation.

S Sliding step between consecutive windows for time series data.

N_w Total number of sliding windows derived from the time series data.

d_{embed} Embedding dimension for the Temporal-Text Embedder.

d_{model} Model dimension used in the attention mechanism for processing both time series and textual data.

d_{ff} Feedforward layer dimension within the Temporal-Text Embedder.

IC Information Coefficient, measuring the correlation between predicted and target values.

$\mathrm{IC}_{\mathrm{target}}$ Target Information Coefficient used in FinLoss.

λ Regularization term in FinLoss for penalizing negative IC.

ICIR Information Coefficient Information Ratio, defined as the ratio of mean IC to its standard deviation over a specific time window.

$\mathrm{ICIR}_{\mathrm{target}}$ Target Information Coefficient Information Ratio, representing the desired stability of IC over time, often used as a threshold for optimizing the trade-off between model performance and stability.

BIBLIOGRAPHY

[1] H.-J. AHN, K. LEE, K. OH, AND K.-H. KIM, *Foreign ownership and stock market liquidity in korea*, Journal of Financial Markets, 12 (2009), pp. 659–676.

[2] A. ANG AND J. CHEN, *Asymmetric correlations of equity portfolios*, Journal of Financial Economics, 63 (2002), pp. 443–494.

[3] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).

[4] L. BAI, L. YAO, C. LI, X. WANG, AND C. WANG, *Adaptive graph convolutional recurrent network for traffic forecasting*, Advances in neural information processing systems, 33 (2020), pp. 17804–17815.

[5] S. M. BARTRAM AND G. DUFEY, *International portfolio investment: Theory, evidence, and institutional framework*, Financial Markets, Institutions & Instruments, 10 (2001), pp. 85–155.

[6] T. BOLLERSLEV, *Generalized autoregressive conditional heteroskedasticity*, Journal of econometrics, 31 (1986), pp. 307–327.

[7] H. BOURLARD AND Y. KAMP, *Auto-association by multilayer perceptrons and singular value decomposition*, Biological cybernetics, 59 (1988), pp. 291–294.

[8] G. E. BOX, G. M. JENKINS, G. C. REINSEL, AND G. M. LJUNG, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015.

[9] D. CAO, Y. WANG, J. DUAN, C. ZHANG, X. ZHU, C. HUANG, Y. TONG, B. XU, J. BAI, J. TONG, ET AL., *Spectral temporal graph neural network for multivariate time-series forecasting*, Advances in neural information processing systems, 33 (2020), pp. 17766–17778.

[10] L. CAO, *AI in finance: Challenges, techniques, and opportunities*, 55, pp. 64:1–64:38.

[11] ——, *Coupling learning of complex interactions*, Information Processing & Management, 51 (2015), pp. 167–186.

[12] ——, *Ai in finance: A review*, Available at SSRN 3647625, (2020).

[13] ——, *Ai in finance: challenges, techniques, and opportunities*, ACM Computing Surveys, 55 (2022), pp. 1–38.

[14] L. Cao, Q. Yang, and P. S. Yu, *Data science and ai in fintech: An overview*, International Journal of Data Science and Analytics, 12 (2021), pp. 81–99.

[15] P. Carr and D. Madan, *Option valuation using the fast fourier transform*, Journal of computational finance, 2 (1999), pp. 61–73.

[16] D. Cheng, Z. Niu, and Y. Zhang, *Contagious chain risk rating for networked-guarantee loans*, in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2715–2723.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078, (2014).

[18] J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex fourier series*, Mathematics of computation, 19 (1965), pp. 297–301.

[19] J. G. De Gooijer and R. J. Hyndman, *25 years of time series forecasting*, International journal of forecasting, 22 (2006), pp. 443–473.

[20] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, Advances in Neural Information Processing Systems, 36 (2024).

[21] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, *Hierarchical multi-scale gaussian transformer for stock movement prediction*, in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, pp. 4640–4646.

[22] M. Dong, H. Huang, and L. Cao, *Can llms serve as time series anomaly detectors?*, 2024.

[23] K. DRAGOMIRETSKIY AND D. ZOSSO, *Variational mode decomposition*, IEEE transactions on signal processing, 62 (2013), pp. 531–544.

[24] Y. DUAN, L. WANG, Q. ZHANG, AND J. LI, *Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4468–4476.

[25] E. F. FAMA AND K. R. FRENCH, *The cross-section of expected stock returns*, the Journal of Finance, 47 (1992), pp. 427–465.

[26] D. FAN, H. SUN, J. YAO, K. ZHANG, X. YAN, AND Z. SUN, *Well production forecasting based on arima-lstm model considering manual operations*, Energy, 220 (2021), p. 119708.

[27] F. FENG, H. CHEN, X. HE, J. DING, M. SUN, AND T.-S. CHUA, *Enhancing stock movement prediction with adversarial training*, arXiv preprint arXiv:1810.09936, (2018).

[28] F. FENG, X. HE, X. WANG, C. LUO, Y. LIU, AND T.-S. CHUA, *Temporal relational ranking for stock prediction*, ACM Transactions on Information Systems, 37 (2019), pp. 1–30.

[29] J. FLEMING, *The quality of market volatility forecasts implied by s&p 100 index option prices*, Journal of empirical finance, 5 (1998), pp. 317–345.

[30] W. GE, P. LALBAKHSH, L. ISAI, A. LENSKIY, AND H. SUOMINEN, *Neural network–based financial volatility forecasting: A systematic review*, ACM Computing Surveys (CSUR), 55 (2022), pp. 1–30.

[31] H. GUNDUZ, *An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination*, Financial innovation, 7 (2021), p. 28.

[32] J. D. HAMILTON, *Time series analysis*, Princeton university press, 2020.

[33] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[34] G. E. HINTON AND R. R. SALAKHUTDINOV, *Reducing the dimensionality of data with neural networks*, science, 313 (2006), pp. 504–507.

[35] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[36] E. HOSEINZADE AND S. HARATIZADEH, *Cnnpred: Cnn-based stock market prediction using a diverse set of variables*, Expert Systems with Applications, 129 (2019), pp. 273–285.

[37] E. J. HU, Y. SHEN, P. WALLIS, Z. ALLEN-ZHU, Y. LI, S. WANG, L. WANG, AND W. CHEN, *Lora: Low-rank adaptation of large language models*, arXiv preprint arXiv:2106.09685, (2021).

[38] N. E. HUANG, Z. SHEN, S. R. LONG, M. C. WU, H. H. SHIH, Q. ZHENG, N.-C. YEN, C. C. TUNG, AND H. H. LIU, *The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis*, Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 454 (1998), pp. 903–995.

[39] T. T. HUYNH, M. H. NGUYEN, T. T. NGUYEN, P. L. NGUYEN, M. WEIDLICH, Q. V. H. NGUYEN, AND K. ABERER, *Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction*, in Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 850–858.

[40] M. JIN, S. WANG, L. MA, Z. CHU, J. Y. ZHANG, X. SHI, P.-Y. CHEN, Y. LIANG, Y.-F. LI, S. PAN, AND Q. WEN, *Time-LLM: Time series forecasting by reprogramming large language models*.

[41] Y. KATZNELSON, *An introduction to harmonic analysis*, Cambridge University Press, 2004.

[42] H. Y. KIM AND C. H. WON, *Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models*, Expert Systems with Applications, 103 (2018), pp. 25–37.

[43] R. KIM, C. H. SO, M. JEONG, S. LEE, J. KIM, AND J. KANG, *Hats: A hierarchical graph attention network for stock movement prediction*, arXiv preprint arXiv:1908.07999, (2019).

[44] T. KIM AND H. Y. KIM, *Forecasting stock prices with a feature fusion lstm-cnn model using different representations of the same data*, PloS one, 14 (2019), p. e0212320.

[45] T. KIM, J. KIM, Y. TAE, C. PARK, J.-H. CHOI, AND J. CHOO, *Reversible instance normalization for accurate time-series forecasting against distribution shift*, in International Conference on Learning Representations, 2021.

[46] D. P. KINGMA AND M. WELLING, *Auto-encoding variational bayes*, arXiv, (2013).

[47] C. LEA, M. D. FLYNN, R. VIDAL, A. REITER, AND G. D. HAGER, *Temporal convolutional networks for action segmentation and detection*, in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 156–165.

[48] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[49] P. LEWIS, E. PEREZ, A. PIKTUS, F. PETRONI, V. KARPUKHIN, N. GOYAL, H. KÜTTLER, M. LEWIS, W.-T. YIH, T. ROCKTÄSCHEL, ET AL., *Retrieval-augmented generation for knowledge-intensive nlp tasks*, Advances in Neural Information Processing Systems, 33 (2020), pp. 9459–9474.

[50] S. LI, X. JIN, Y. XUAN, X. ZHOU, W. CHEN, Y.-X. WANG, AND X. YAN, *Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting*, Advances in neural information processing systems, 32 (2019).

[51] B. LIM, S. Ö. ARIK, N. LOEFF, AND T. PFISTER, *Temporal fusion transformers for interpretable multi-horizon time series forecasting*, International Journal of Forecasting, 37 (2021), pp. 1748–1764.

[52] P. LIU, H. GUO, T. DAI, N. LI, J. BAO, X. REN, Y. JIANG, AND S. XIA, *Calf: Aligning llms for time series forecasting via cross-modal fine-tuning*, arXiv preprint arXiv:2403.07300, (2024).

[53] S. G. MALLAT, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE transactions on pattern analysis and machine intelligence, 11 (1989), pp. 674–693.

[54] J. MASCI, U. MEIER, D. CIREŞAN, AND J. SCHMIDHUBER, *Stacked convolutional auto-encoders for hierarchical feature extraction*, in Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21, Springer, 2011, pp. 52–59.

[55] L. MYERS AND M. J. SIROIS, *Spearman correlation coefficients, differences between*, Encyclopedia of statistical sciences, 12 (2004).

[56] A. NG ET AL., *Sparse autoencoder*, CS294A Lecture notes, 72 (2011), pp. 1–19.

[57] Y. NIE, N. H. NGUYEN, P. SINTHONG, AND J. KALAGNANAM, *A time series is worth 64 words: Long-term forecasting with transformers*, arXiv preprint arXiv:2211.14730, (2022).

[58] A. V. D. OORD, S. DIELEMAN, H. ZEN, K. SIMONYAN, O. VINYALS, A. GRAVES, N. KALCHBRENNER, A. SENIOR, AND K. KAVUKCUOGLU, *Wavenet: A generative model for raw audio*, arXiv preprint arXiv:1609.03499, (2016).

[59] OPENAI, *Gpt-4 technical report*, 2023.

[60] A. J. PATTON, *On the out-of-sample importance of skewness and asymmetric dependence for asset allocation*, Journal of Financial Econometrics, 2 (2004), pp. 130–168.

[61] D. E. RUMELHART, G. E. HINTON, AND R. J. WILLIAMS, *Learning representations by back-propagating errors*, nature, 323 (1986), pp. 533–536.

[62] R. SAWHNEY, S. AGARWAL, A. WADHWA, T. DERR, AND R. R. SHAH, *Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach*, 35, pp. 497–504.
Number: 1.

[63] ——, *Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 497–504.

[64] R. SAWHNEY, S. AGARWAL, A. WADHWA, AND R. SHAH, *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, 2020, pp. 8415–8426.

[65] R. SAWHNEY, S. AGARWAL, A. WADHWA, AND R. R. SHAH, *Spatiotemporal hypergraph convolution network for stock movement forecasting*, in 2020 IEEE International Conference on Data Mining, IEEE, 2020, pp. 482–491.

[66] S. SIAMI-NAMINI, N. TAVAKOLI, AND A. S. NAMIN, *A comparison of arima and lstm in forecasting time series*, in 2018 17th IEEE international conference on machine learning and applications (ICMLA), IEEE, 2018, pp. 1394–1401.

[67] H. S. SIM, H. I. KIM, J. J. AHN, ET AL., *Is deep learning for image recognition applicable to stock market prediction?*, Complexity, 2019 (2019).

[68] N. SRIVASTAVA, E. MANSIMOV, AND R. SALAKHUDINOV, *Unsupervised learning of video representations using lstms*, in International conference on machine learning, PMLR, 2015, pp. 843–852.

[69] C. TIAN, T. NIU, AND W. WEI, *Volatility index prediction based on a hybrid deep learning system with multi-objective optimization and mode decomposition*, Expert Systems with Applications, 213 (2023), p. 119184.

[70] H. TOUVRON, T. LAVRIL, G. IZACARD, X. MARTINET, M.-A. LACHAUX, T. LACROIX, B. ROZIÈRE, N. GOYAL, E. HAMBRO, F. AZHAR, ET AL., *Llama: Open and efficient foundation language models*, arXiv preprint arXiv:2302.13971, (2023).

[71] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, Advances in neural information processing systems, 30 (2017).

[72] P. VINCENT, H. LAROCHELLE, Y. BENGIO, AND P.-A. MANZAGOL, *Extracting and composing robust features with denoising autoencoders*, in Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.

[73] G. WANG, L. CAO, H. ZHAO, Q. LIU, AND E. CHEN, *Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty*, 35, pp. 4418–4426.

[74] ——, *Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 4418–4426.

[75] Y. Wang, A. Smola, D. Maddix, J. Gasthaus, D. Foster, and T. Januschowski, *Deep factors for forecasting*, in International conference on machine learning, PMLR, 2019, pp. 6607–6617.

[76] H. Wu, J. Xu, J. Wang, and M. Long, *Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting*, Advances in Neural Information Processing Systems, 34 (2021), pp. 22419–22430.

[77] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, *Connecting the dots: Multivariate time series forecasting with graph neural networks*, in Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 753–763.

[78] W. Xu, W. Liu, L. Wang, Y. Xia, J. Bian, J. Yin, and T.-Y. Liu, *Hist: A graph-based framework for stock trend forecasting via mining concept-oriented shared information*, arXiv preprint arXiv:2110.13716, (2021).

[79] H. Yang, X.-Y. Liu, and C. D. Wang, *Fingpt: Open-source financial large language models*, arXiv preprint arXiv:2306.06031, (2023).

[80] Y. Yang and L. Cao, *Mtsnet: Deep probabilistic cross-multivariate time series modeling with external factors for covid-19*, in 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, 2023, pp. 1–10.

[81] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, *Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective*, in Thirty-seventh Conference on Neural Information Processing Systems, 2023.

[82] J. Zhan, J. Dai, J. Ye, Y. Zhou, D. Zhang, Z. Liu, X. Zhang, R. Yuan, G. Zhang, L. Li, et al., *Anygpt: Unified multimodal llm with discrete sequence modeling*, arXiv preprint arXiv:2402.12226, (2024).

[83] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, *Informer: Beyond efficient transformer for long sequence time-series forecasting*, in Proceedings of the AAAI conference on artificial intelligence, vol. 35, 2021, pp. 11106–11115.

[84] J. Zhou, S. Wang, and Y. Ou, *Fourier graph convolution transformer for financial multivariate time series forecasting*, in 2024 International Joint Conference on Neural Networks (IJCNN), IEEE, 2024, pp. 1–8.

[85] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, *Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting*, in International Conference on Machine Learning, PMLR, 2022, pp. 27268–27286.

[86] Q. Zhu, F. Zhang, S. Liu, Y. Wu, and L. Wang, *A hybrid vmd–bigru model for rubber futures time series forecasting*, Applied Soft Computing, 84 (2019), p. 105739.