Fourier Graph Convolution Transformer for Financial Multivariate Time Series Forecasting

Junxian Zhou, Shoujin Wang, Yuming Ou
University of Technology Sydney

***Abstract*—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 paper, 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://anonymous.4open.science/r/FreTransformer**

***Index Terms*—Multivariate Time Series, Financial Time Series Forecasting, Transformer, Frequency domain, Graph.**

I. 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 [1], [2]. 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 [3], [4]. 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 [5]. 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) [6] and Auto-Regression (AR) [3], 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 [7], HIST [8] and ESTIMATE [5] 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.

II. Related Work

Graph deep models for financial multivariate time series forecasting. Fin-MTS has adopted Graph Neural Networks (GNN) because of their superior performance in modelling the complex structural representations among the different assets [5], [8]–[11]. Most GNNs adopt a pre-fixed graph structure to capture the correlations and representations among assets, as exemplified by models like HIST [8]. Some models, such as MAN-SF [12] and TRACER [13], 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 [5], [8], [12] 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 [14] to conduct a novel study of capturing intra- and inter-series information within the Fourier Space harmonic signals.

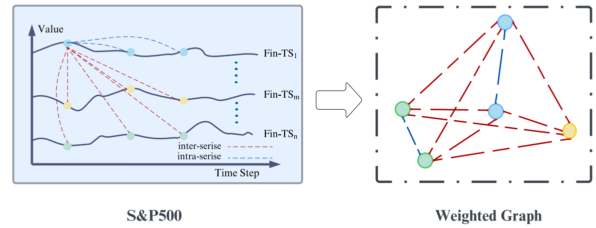


Fig. 1. Illustration of the weighted graph with n distinct Fin-TS input. Fin-TS denotes the Financial Time Series. Each feature within every time step of each Fin-TS is represented as a node in a weighted graph.

Frequency based complex value Transformer for multivariate time series forecasting. Many Multivariate Time Series deep learning forecasting models harmonized with the frequency domain concept with complex value to improve the performance [15]–[18]. For example, SFM [19] enhances the vanilla LSTM by decomposing its latent space with the Discrete Fourier Transform (DFT), leading to more accurate stock price predictions. Recently, FEDformer [17] has been an innovative network using frequency domain knowledge with the low-rank approximation and Discrete Fourier Transform-based attention mechanism. Autoformer [16], 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. Several newest models [18], [20] aim to understand intra-series correlations in the frequency domain by analysing amplitude values after applying the FFT. They use these amplitude values to decompose the original series into several distinct periods. However, there is still a gap in capturing intra- and inter-series dependency in the frequency domain by current frequency-based Transformers. They only utilise the constant value weighted matrix, whereas the MTS converted in Fourier Space includes two types of information: amplitude and phase. Without the proper learnable weighted matrix after the FFT, frequency-based Transformer models will lose the intra- and inter-series details. Therefore, intra- and inter-series information is incomplete in current frequency Transformer models. This paper proposes the FreTransformer, a model designed to study the information from both intra- and inter-series information without the aforementioned loss. It addresses the challenge of Transformer model learning of intra- and inter-series features in the frequency domain.

III. Fourier Graph Convolution Transformer

A. The FreTransformer Architecture

This paper introduces a thoroughly frequency-based Transformer with FGCN for modelling the Fin-MTS. Before proceeding to the FFT, the Fin-MTS will be considered as a fully connected weighted graph, detailed in Section III-B. The lossless domain transformation will be shown in Section III-C to prove the graph meaning in the frequency-based MTS

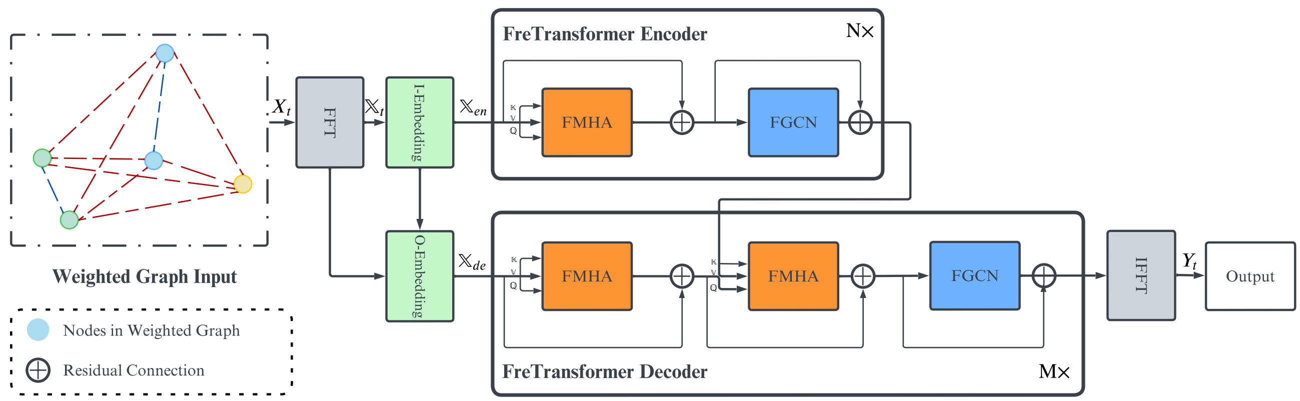


Fig. 2. FreTransformer Architecture. $\mathbb{X}_t$ discloses the hidden periodicities. The FGCN effectively captures both the intra- and inter-series dependencies as well as hidden periodicities. The Fourier Multi-Head Attention Mechanism (FHMA) is a naïve variant of the basic Multi-Head Attention Mechanism.

forecasting. Section III-D specifics FGCN, which is the core module for the FreTransformer to learn both intra- and inter-series features in the frequency domain. The FreTransformer, a sequence-to-sequence model with an input-encoder-decoder-output framework, is shown in Fig 2.

B. Prior Fully Connected Weighted Graph Structure

The prior graph structure, a novel approach in Fin-MTS forecasting scenarios [14], [21]–[23] leads to the following definitions for forming a Fully Connected Weighted Graph.

Definition 1 (Graph Fin-MTS forecasting problem). Given a 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 $\hat{\boldsymbol{X}}_{\boldsymbol{t}} = [\boldsymbol{x_{t-T+1}}, \ldots, \boldsymbol{x_t}]^T \in \mathbb{R}^{T \times N}$ as the lag correlation feature. Fin-MTS forecasting problem can be defined as to predict 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 $\hat{\boldsymbol{X}}_{\hat{\boldsymbol{t}}} = [\boldsymbol{x_{\hat{t}-T+1}}, \ldots, \boldsymbol{x_{\hat{t}}}]^T \in \mathbb{R}^{\hat{t} \times N}$. Formulas lead to the Fin-MTS forecasting procedure as follows:

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

where $\hat{\boldsymbol{Y}}_{\boldsymbol{t}}$ 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. After transforming to a weighted graph, the original task can be reformulated as:

$$\hat{\boldsymbol{Y}}_{\boldsymbol{t}} = \boldsymbol{f}_{\theta,\theta_g}(\hat{\boldsymbol{X}}_{\boldsymbol{t}}) = \boldsymbol{f}_{\theta,\theta_g} \left[\boldsymbol{x_{t-T+1}}, \ldots, \boldsymbol{x_t}\right]^T \in \mathbb{R}^{\hat{\tau} \times N}. \quad (2)$$

Equation 2 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.

C. Domain Transformation/Reversal

We have the fundamental lossless tool, Fourier Transform, to transfer the time domain data to the frequency domain. Due to the Fin-MTS being the discrete data, we have FFT [24] to decompose it to capture the cyclical trend patterns within the data advantageously. Thus, we have the discrete input $\boldsymbol{X_t}$ convert into frequency domain:

$$\begin{aligned}
\mathbb{X}_t &= \mathrm{FFT}(\boldsymbol{X_t}) = A(\boldsymbol{X_t}) e^{j\Phi(\boldsymbol{X_t})} \\
&= \left[\sum_{n=0}^{N-1} X_t[t,n] \cdot e^{-\frac{2\pi i}{N}kn}\right]_{k=0}^{N-1} \\
&= \mathrm{Re}\left(\sum_{n=0}^{N-1} X_t[t,n] \cdot e^{-\frac{2\pi i}{N}kn}\right) \\
&+ i \cdot \mathrm{Im}\left(\sum_{n=0}^{N-1} X_t[t,n] \cdot e^{-\frac{2\pi i}{N}kn}\right), \\
&\forall t \in \{1, 2, \ldots, T\},
\end{aligned} \quad (3)$$

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 $\mathbb{X}_t = \mathit{Re}(\mathbb{X}_t) + i \cdot \mathit{Im}(\mathbb{X}_t)$, where $\mathit{Re}(\mathbb{X}_t)$ and $\mathit{Im}(\mathbb{X}_t)$ separately rewrite the real part and imaginary part in the Equation 3.In subsequent sections, for simplicity, we will refer to Equation 3 as `FFT` and Equation 4 as `iFFT`. As one of the most critical recent algorithms, FFT has the lossless reversal method iFFT [24], which is shown as:

$$\begin{aligned}
\boldsymbol{X_t} &= \mathrm{IFFT}(\mathbb{X}_t) \\
&= \left[\frac{1}{N}\sum_{k=0}^{N-1} \mathbb{X}_t[t,k] \cdot e^{\frac{2\pi i}{N}kn}\right]_{n=0}^{N-1} \\
&= \frac{1}{N}\left(\mathrm{Re}\left(\sum_{k=0}^{N-1} \mathbb{X}_t[t,k] \cdot e^{\frac{2\pi i}{N}kn}\right)\right) \\
&+ i \cdot \frac{1}{N}\left(\mathrm{Im}\left(\sum_{k=0}^{N-1} \mathbb{X}_t[t,k] \cdot e^{\frac{2\pi i}{N}kn}\right)\right), \\
&\forall t \in \{1, 2, \ldots, T\}.
\end{aligned} \quad (4)$$

D. FGCN

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 [14].

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) [25] 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 [26] can be rewritten as:

$$\text{FFT}(X)\text{FFT}(\kappa) = \text{FFT}((X * \kappa)[i][j]), \tag{5}$$

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 Equations (3) to (5) as:

$$\begin{aligned}\text{FGCN}_W(X, A) &= \sigma(\sum_{l=0}^{L}(\text{FFT}(AXW) + \mathbb{B}))\\ \text{FGCN}_{\mathfrak{K}}(\mathbb{X}) &= \sigma(\sum_{l=0}^{L}(\mathbb{X} \times \mathfrak{K}[i][j] + \mathbb{B})).\end{aligned} \tag{6}$$

Proof. The proof demonstrates the equivalence in Equation 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:

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

where we arrive at the equivalence expression $\text{FGCN}_W(X, A) = \text{FGCN}_{\mathfrak{K}}(\mathbb{X})$ in the Equation 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:

$$\begin{aligned}&\mathbb{X} \times \mathfrak{K}[i][j] + \mathbb{B}\\ =&(Re(\mathbb{X}) + i \cdot Im(\mathbb{X}))\cdot\\ &(Re(\mathfrak{K})[i][j] + i \cdot Im(\mathfrak{K}[i][j])) + \mathbb{B}\\ =&\,(Re(\mathbb{X})Re(\mathfrak{K})[i][j] - Im(\mathbb{X})Im(\mathfrak{K}[i][j]) + Re(\mathbb{B}))+\\ &i(Re(\mathbb{X})Im(\mathfrak{K}[i][j]) + Im(\mathbb{X})Re(\mathfrak{K})[i][j] + Im(\mathbb{B})).\end{aligned} \tag{7}$$

E. *FreTransformer*

We have innovated the Transformer [27] model by integrating it with an FGCN, creating a frequency-based deep learning structure named FreTransformer, as shown in Fig. 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) [27] to learn the intra-series features with phase in the frequency domain. The inputs are formulated as follows:

$$\begin{aligned}\mathbb{X}_{en} &= \text{FGCN}_{\mathfrak{K}}(\mathbb{X}) + \text{PE}(\mathbb{X})\\ \mathbb{X}_{de} &= \text{Concat}(\mathbb{X}_{en}, \mathbb{X}).\end{aligned} \tag{8}$$

Fourier Multi-Head Attention Mechanism. The Fourier Multi-Head Attention (FMHA) is a variation of the original attention mechanism [27]. 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:

$$\mathbb{Q} = \text{Complex}(\text{Linear}(Re(\mathbb{X}_{en})), \text{Linear}(Im(\mathbb{X}_{en})). \tag{9}$$

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

$$\begin{aligned}\text{FMHA}(\mathbb{Q}, \mathbb{K}, \mathbb{V}) = \text{Complex}(&\text{Softmax}(Re(\frac{\mathbb{Q}\mathbb{K}^T}{\sqrt{d_{\mathbb{K}}}})),\\ &\text{Softmax}(Im(\frac{\mathbb{Q}\mathbb{K}^T}{\sqrt{d_{\mathbb{K}}}})) \cdot \mathbb{V}.\end{aligned} \tag{10}$$

Encoder/Decoder. As illustrated in Fig. 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$(x + \text{Sublayer}(x))$ which is a residual link design. The equations are specified as:

$$\begin{aligned}\mathbb{X}_{en}^{l,1}, \_ &= \text{AddNorm}(\text{FMHA}(\mathbb{X}_{en}^{l-1,1}) + \mathbb{X}_{en}^{l-1,1})\\ \mathbb{X}_{en}^{l,2}, \_ &= \text{AddNorm}(\text{FGCN}(\mathbb{X}_{en}^{l,1}) + \mathbb{X}_{en}^{l,1}),\end{aligned} \tag{11}$$

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:

$$\begin{aligned}\mathbb{X}_{de}^{l,1} &= \text{AddNorm}(\text{FMHA}(\mathbb{X}_{de}^{l-1,1}) + \mathbb{X}_{de}^{l-1,1})\\ \mathbb{X}_{de}^{l,2} &= \text{AddNorm}(\text{FMHA}(\mathbb{X}_{de}^{l,1}, \mathbb{X}_{en}^{N}) + \mathbb{X}_{de}^{l,1})\\ \mathbb{X}_{de}^{l,3} &= \text{AddNorm}(\text{FGCN}(\mathbb{X}_{de}^{l,2}) + \mathbb{X}_{de}^{l,2}),\end{aligned} \tag{12}$$

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 $\mathbb{X}_{de}^{l}$.

Fig. 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.

IV. Experiments

This section details the evaluation process for our FreTransformer comprehensive analysis of four 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.

A. Benchmarks

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

- LSTM [4], 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 [28], 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 [8], 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 [5], 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 I,II.

B. 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 trading days) and split the data into 12 phases [5], [8] due to the markets period with different representations, volatility, trading volume, and markets sentiment in Fig 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.

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.

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

TABLE I
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_{std}	0.022	0.028	0.022	**0.008**	0.057	0.018*
IC_{mean}	-0.006	-0.007	0.003	0.003	0.045*	**0.113**
$Diff_{IC_m}$	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**
$Diff_{ICIR}$	0.000	+0.050	+0.421	+0.608	+1.059*	**+6.572**

TABLE II
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_{std}	0.251	0.267	0.228	0.181	0.053*	**0.034**
R_IC_{mean}	-0.101	-0.096	-0.007	0.080*	0.035	**0.137**
$Diff_{R\_IC_m}$	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**
$Diff_{R\_ICIR}$	0.000	+0.043	+0.373	+0.841	+1.073 *	**+4.458**

- *Rank information ratio based IC (Rank_ICIR):* is the information ratio of computed by average Rank_IC and standard deviation of the rank_IC [29].

Reproducibility environment. 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.

C. Performance

The financial time series forecasting performance on the S&P500 dataset is detailed in Table I,II, with baselines comparative data originating from [5]. In terms of `IC` and `Rank_IC` Metrics, our FreTransformer outperformed the current deep financial models [27] 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`.

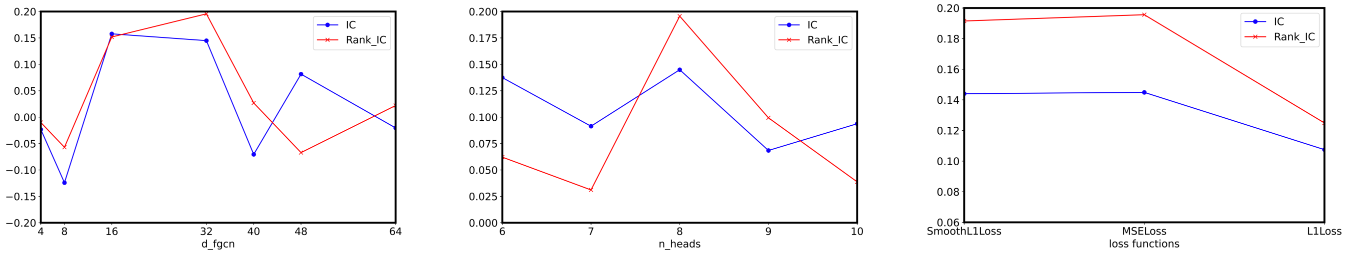
TABLE III
OVERALL ACCURACY, IC, RANK_IC.

	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

D. Ablation Study

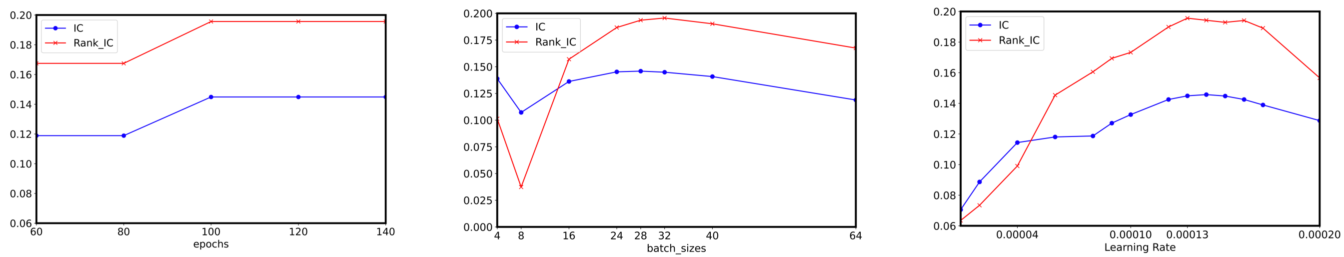
To address the Research Question (RQ2), we assessed the significance of each component within our model by developing four distinct variants: `(FT-1)` This variant eliminates the FGCN operators in the encoder layers within FreTransformer. `(FT-2)` This variant eliminates the FGCN operators in the decoder layers within FreTransformer. `(FT-3)` This variant

[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.



(a) FGCN Dimensions (b) Multi-head Attention (c) Loss Functions

Fig. 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.



(a) Epochs (b) Batch Sizes (c) Learning Rate

Fig. 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}.

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

E. 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.

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 requirement, and the drift of the concept or data. To independently evaluate and 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 (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 (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 (c).

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. 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. 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.5 (c) shows the study results and reviews.

V. Conclusion

In this paper, we explore a novel model, the FreTransformer, which overcomes the aforementioned hidden periodic problem. The model utilises a prior graph with an FGCN to map the Fin-MTS into the frequency domain. This approach addresses the limitations associated with forecasting loss in Fin-MTS. Extensive experiments demonstrate that the FreTransformer achieves state-of-the-art performance. Additionally, the structure of the FreTransformer exhibits strong capabilities in capturing both intra- and inter-series dependencies.

References

[1] L. Cao, "Ai in finance: challenges, techniques, and opportunities," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1–38, 2022.

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

[3] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] 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.

[6] J. D. Hamilton, *Time series analysis*. Princeton university press, 2020.

[7] 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*, vol. 37, no. 2, pp. 1–30, 2019.

[8] 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.

[9] 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.

[10] 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.

[11] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 497–504.

[12] 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.

[13] 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.

[14] 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.

[15] K. Yi, Q. Zhang, L. Cao, S. Wang, G. Long, L. Hu, H. He, Z. Niu, W. Fan, and H. Xiong, "A survey on deep learning based time series analysis with frequency transformation," *CoRR, abs/2302.02173*, 2023.

[16] 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*, vol. 34, pp. 22 419–22 430, 2021.

[17] 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. 27 268–27 286.

[18] T. Zhou, Z. Ma, Q. Wen, L. Sun, T. Yao, W. Yin, R. Jin *et al.*, "Film: Frequency improved legendre memory model for long-term time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 677–12 690, 2022.

[19] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 2141–2149.

[20] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[21] 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*, vol. 33, pp. 17 766–17 778, 2020.

[22] 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*, vol. 33, pp. 17 804–17 815, 2020.

[23] 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.

[24] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] Y. Katznelson, *An introduction to harmonic analysis*. Cambridge University Press, 2004.

[27] 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*, vol. 30, 2017.

[28] 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.

[29] L. Myers and M. J. Sirois, "Spearman correlation coefficients, differences between," *Encyclopedia of statistical sciences*, vol. 12, 2004.