Pattern Recognition 122 (2022) 108209

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Graph-based stock correlation and prediction for high-frequency trading systems

Tao Yin [a], Chenzhengyi Liu [b], Fangyu Ding [b], Ziming Feng [c], Bo Yuan [d], Ning Zhang [b,*]

[a] Department of Computer Science and Engineering, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, China
[b] Department of Computer Science and Engineering, and MOE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, China
[c] China Merchants Bank Credit Card Center, No. 385, Ruichang Road, Shanghai, China
[d] Department of Computer Science and Engineering, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, China

ARTICLE INFO

Article history:
Received 30 November 2020
Revised 22 June 2021
Accepted 27 July 2021
Available online 11 August 2021

Keywords:
High-frequency trading
Graph attention long short-term memory (GALSTM)
Hawkes processes
Portfolio management

ABSTRACT

In this paper, we have implemented a high-frequency quantitative system that can obtain stable returns for the Chinese A-share market, which has been running for more than 3 months (from March 27, 2020 to June 30, 2020) with the expected results. A number of rules and barriers exist in the Chinese A-share market such as trading restrictions and high fees, as well as scarce and expensive hedging tools. It is difficult to achieve stable absolute returns in such a market. Stock correlation analysis and price prediction play an important role to achieve any profitable trading. The portfolio management and subsequent trading decisions highly depend on the results of stock correlation analysis and price prediction. However, it is nontrivial to analyze and predict any stocks, being time-varying and affected by unlimited factors in a given market. Traditional methods only take some certain factors into consideration but ignore others that may be changed dynamically. In this paper, we propose a novel machine learning model named Graph Attention Long Short-Term Memory (GALSTM) to learn the correlations between stocks and predict their future prices automatically. First, a multi-Hawkes Process is used to initial a correlation graph between stocks. This procedure provides a good training start as the multi-Hawkes Processes will be studied on the most saint feature fluctuations with any correlations being statistically significant. Then an attention-based LSTM is built to learn the weighting matrix underlying the dynamic graph. In addition, we also build matching data process plus portfolio management modules to form a complete system. The proposed GALSTM enables us to expand the scope of stock selection under the premise of controlling risks with limited hedging tools in the A-share market, thereby effectively increasing high-frequency excess returns. We then construct a long and short positions combination, select long positions in the A shares of the entire market, and use stock index futures to short. With GALSTM model, the products managed by our fully automatic quantitative trading system achieved an absolute annual return rate of 44.71% and the standard deviation of daily returns is only 0.42% in three months of operation. Only 1 week loss in 13 weeks of running time.

1. Introduction

With the continuous improvement from both of the computer hardware and AI technology, quantitative investment as a large investment category is growing rapidly, especially in the Chinese market. In the initial stage of quantitative investment, computers were mainly used to automatically process peoples ideas in investment. Today, the more promising direction is to build a fully automatic trading system based on data, statistics, and artificial intelligence to achieve stable profits. The advantages of the fully automatic trading system are fast execution, large capacity, and good risk control. The advantage of the artificial intelligence method is that it does not require a special in-depth domain knowledge of each stock. The model is easy to migrate, and more strategies can be developed per unit time to obtain greater profits.

High-frequency trading (HFT) [1] is a trading method that uses powerful computer programs to execute a large amount of orders in a short period of time. Sophisticated algorithms are used to analyze markets and execute orders. A well built HFT systems

* Corresponding author.
E-mail addresses: yintao_sj@sjtu.edu.cn (T. Yin), liuchenzhengyi@sjtu.edu.cn (C. Liu), arthur_99@sjtu.edu.cn (F. Ding), zimingfzm@cmbchina.com (Z. Feng), boyuan@sjtu.edu.cn (B. Yuan), ningz@sjtu.edu.cn (N. Zhang).

https://doi.org/10.1016/j.patcog.2021.108209
0031-3203/

can make profits at a very low risk, which is similar to fixed income.[1] However, the overall capacity of HFT in a certain market is rather limited. As increasingly more companies and organizations are competing in this limited capacity, the competition is more and more intensive. It always demand leading technologies, including mathmatical or algorithm models, for the companies to achieve profit.

In this paper, we focus on the Chinese A-share market. In this market, there are some difficulties in applying high-frequency quantitative trading to obtain stable absolute returns. For example, the trading restrictions, high fees, scarce and expensive hedging tools. Specifically, first the $T + 1$ trading rules: that is, the stocks bought on the same day cannot be sold. It is worth noting that buying is allowed after the stock is sold in one day. In this case, we can only achieve high-frequency trading by holding a long position in the stock and using a short position tool as a hedge. Based on this, we can sell yesterdays stock position and buy the corresponding stock. Therefore, if we want to get stable income, hedging is very important. Secondly, fees are relatively high: For a transaction with buy and sell, the costs are usually above 0.12% of the transaction values. Thirdly, lack of hedging: The only official instrument with better liquidity is stock index futures, but there are fewer types of stock index futures. Taking the most commonly used IC contract as an example, which corresponds to the CSI 500 Index.[2] The constituent stocks of CSI 500 have relatively small fluctuations, which means that short-term trading profits are severely restricted. And the average cost of hedging in first half of 2020 is more than 1% a month. In this environment, if we can better analyze the correlations of stocks and predict their future prices, we can not only increase the return of a single stock, but hedge more targets under the similar risk. To achieve the purpose of further improving the funds income and capacity.

To develop a high-frequency trading system that can solve the practical problems of achieving stable profitability in the Chinese A-share market, one need to first build a portfolio and then maximize the profitability in a single order. To build better portfolio, we analyze the correlations between stocks. To maximize the per order profit, it is useful to predict the future price of each stock. The common approaches for stocks' correlation analysis are to classify the stocks based on some certain factors, such as market value factor, industry factor, etc. Disadvantages of this approach are obvious. First the classification is relatively fixed and cannot be adjusted in real time with the market situation changes. Secondly, the factors used can only describe influences from limited aspects, and the comprehensive influences of various factors are not taken into consideration. The price predictions in previous approaches are not as productive, which is related to ignoring the comprehensive correlations between stocks.

In this paper, we propose a novel Graph-based model called Graph Attention Long Short-Term Memory (GALSMT) to capture the correlations between different stocks and predict their future prices. To be specific, GALSTM first adopts the Hawkes Process [2] to extract the correlation features of stocks' events and generate a directed graph named correlation graph. Next, adding stock prices at each time interval to the correlation graph, GALSTM develops an algorithm which allows a peephole LSTM to recurrently process the directed graph and update its edge weights at each time interval as well as output the prediction of future prices. Using GALSTM allows our system more accurately predict the relative price changes between stocks and the absolute changes in individual stock prices. Thus with limited hedging tools, more stocks can be used for trading under similar risk exposures, especially some high-volatility stocks. This lead to the income increasing of a single transaction. The quantitative trading system we actually run is very complicated. In addition to the above method innovations, we have also balanced the use of various classic technologies.

The proposed trading system has run on a fund product registered by the China Fund Industry Association, which is the official institution of China. The performance of the product is public and can be acquired on the relevant website. In the three months of actual operation, our engine has achieved an absolute annual return rate of 44.71% and the standard deviation of daily returns is only 0.42% in three months of operation. Only 1 week loss in 13 weeks of running time. In the experiment part, we present the performance curve and related websites.

The main contributions of this paper are summarized as following:

- We identify that the traditional factor-based risk control methods in high frequency trading have become the bottle-neck for trading systems in Chinese A-share market.
- This paper proposes a novel GALSTM model combined with multi-Hawkes process to learn the correlations between stocks and predict the prices of these stocks.
- We build the corresponding data process module, portfolio management module and strategy implementation module to collaborate with the proposed GALSTM model.
- The proposed system has achieved stable income in real Chinese A-share market for three months.

2. Related work

2.1. Quantitative trading system

As mentioned above, the actual profitable quantitative trading system is a complicated project, which consists a series of interactive modules. Related researches can be categorized according to the different modules they focus on. (1) The data source related works, such as fundamental analysis, specific events [3]. (2) Some specific trading methods, such as arbitrage and pairs trading [4,5]. (3) Researches on trading strategy, such as mean-reverting [6], momentum strategy [7,8], reinforcement learning strategy. (4) Transaction execution models, such as algorithmic trading. (5) Portfolio optimization methods, such as portfolio management [9]. (6) Researches on forecasting problems, such as mining factor [10], machine learning methods to predict price changes. There are some other kinds of works, which we cannot list them all.

2.2. Alpha research

Alpha research is one of the most common research directions in stock quantitative trading. In the field of finance, alpha is the measure of the excess return of an investment over a suitable benchmark, which can be index of a market or an industry. Alphas are fueled by data. Some typical data sources includes: prices and volumes, fundamentals, macroeconomic data, texts (such as Federal Open Market Committee minutes, company filings, papers, journals, news, or social media), multimedia, notably relevant videos or audios [11].

Prices and volumes, fundamentals, and macroeconomic data are relatively easy to standardize. Most of the relevant data is standardized by data service providers and sent to researchers. But text and media are more non-standard, and often the information contained is not clear enough to find alpha. These data need to be analyzed manually, or use some artificial intelligence methods to extract signals and standardize [12].

[1] https://www.investopedia.com/terms/f/fixedincome.asp
[2] http://www.csindex.com.cn/zh-CN/indices/index-detail/000905

In the Chinese A-share market, the proposed method in this paper is usually classified as $T + 0$ (intraday trading) research, not alpha research. The difference is that in essence the source of profit of the proposed method is the short-term price prediction of a single stock. The contribution of this paper is to adapt to the trading rules of the Chinese market, increase returns and control risks. The proposed method does not pursue the excess return of the investment over the benchmark.

2.3. AI technologies in quantitative trading

AI technologies are mainly applied in two directions: trading strategy and price prediction.

At present, some progress have been made in trading strategies. For example, by using the reinforcement leaning improved trading method, 10 times increase have been achieved in portfolio over 17 years no transaction cost in Pendharkar and Cusatis [13]. By combining the recurrent reinforcement learning and the genetic Algorithms, Zhang and Maringer [14] obtained 1 Sharpe ratio on average. Despite the progresses have been made, these methods are still deficiencies. For example, some practical factors in real markets such as transaction costs are not considered, and the Sharpe ratio is not high enough. These existing systems are not good enough for real trade.

Various kinds of machine learning techniques have been using for price prediction. The decision tree based methods have been proposed in Ren et al. [15] and Tsai et al. [16]. They claimed to achieve accuracy of 82% [15] and 65.41% [16], respectively. The work [17,18] proposed prediction models based on support vector machine (SVM) techniques, and they reached the prediction accuracy 61.73% and 96.46%. The work [19,20] have built their forecasting system based on neural networks, and reported the prediction accuracy of 59.38% and 87.5%. The Bayesian Networks have also been used in the price prediction models [21–23] reached 92% 76% 86% and another method [24] proposes a cost-sensitive deep forest for price prediction with a classification method.

Due to different forecasting instruments and different forecasting targets, it is difficult to make a unified comparison for all these trading systems. From a purely forecasting point of view, the accuracy of some results is also very high. However, from the perspective of actual profit, there are still some unresolved problems, especially for high-frequency trading. There is still a considerable distance between these forecasting models and the real transaction systems, which include the probability of transaction, the coordination of trading strategies, and actual order profit expectation after considering the transaction problems.

2.4. Data-driven approaches for quantitative trading

To obtain the correlations between the time series of different stocks' prices, researchers have proposed many methods. The transfer entropy method [25] answers the question from the aspect of information theory. The attention-based models like [26,27] address the time series forecasting problems by finding out the relationships between different time series.

Some temporal point process based prediction models can also be adopted in this task. The work [28] takes a reinforcement learning view to discover the temporal patterns via event sequences clustering. The authors in Yan et al. [29] propose a modern machine learning paradigm to estimate the win-propensity of sales leads over time, which also mainly deals with the transaction data.

Other researches focusing on graph models are also inspiring. Lu et al. [30] proposes a novel unsupervised feature extraction method to extract the similarity relationship between samples in graphs. Hu et al. [31] presents a novel framework which models the neighborhood interaction and aggregation in graph.

Recently, a few works combine the Temporal Point Process (TPP) [32] and Graph Neural Network (GNN) [33] to improve the model capacity and expressiveness in sequence data analysis, which are more relevant to our work. Specifically, Zhou et al. [34] and Li and Zha [35] adopt the Multi-dimensional Hawkes (MHP) process to dig the underlying topology of social networks. In spatio-temporal events prediction problem, Liu et al. [36] designs a graph regularization method to effectively integrate the prior spatial structure into MHP for learning influence matrix between different locations. The work [37] takes a step further and incorporates additional geometric structure in the form of graphs into Hawkes processes by Geometric Hawkes Process. These works do achieve good performance in their specific tasks. However, most of them are still focus on events and thus cannot give accurate prediction for fixed-interval time series like stock price for trading systems.

3. The proposed approach and system

The framework of the proposed system is shown in Fig. 1, which consists of five main components: (1) Data Process; (2) Correlation Analysis and Prediction; (3) Strategy Implementation; (4) Portfolio Management; (5) Trading Platform.

China's A-share market data consists of tick, order, and transaction. First, it is processed by the data processing module to extract all the characteristics of each stock into a *FairPrice* sequence. This enables the system to better capture the patterns of stocks regardless of some human-made price fluctuations in raw dateset. Besides, we choose our *FairPrice* of stocks based on experiment results.

Secondly, the correlations between different stocks are analyzed by processing their *FairPrice* with our proposed GALSTM. The future prices of these stocks are predicted at the same time. Then, the GALSTM, which is designed as a novel graph-based recurrent neural network, helps the system to capture the stocks' correlations more accurately. Thus with limited hedging tools, more stocks can be used for trading under similar risk exposures, especially for some high-volatility stocks.

Next is the strategy implementation step. In this part, the actual transactions are based on the individual stock predictions obtained by the above model. The output of model is used as a signal to trigger the transaction and determine the direction of the transaction. At the same time, the transaction price and volume are determined according to the model output and the high-frequency trading strategies. Then the actual transaction orders can be obtained.

Last, the dynamic long positions are constructed in the portfolio management module. The correlation diagram obtained by the GALSTM model is used to assist in the risk judging for a single stock. Then the expected return of this single stock is estimated via the backtesting. To achieve the goal of stable profitability, the stocks in trading are dynamically changed to balance the returns and risks.

All the above modules compose a training platform, which can be applied in real market. We will introduce these modules in detail in the following sections.

3.1. Data processing

Market data of Chinese A-share is mainly divided into tick, order and trans. The tick data displays snapshots with a frequency of 3 seconds. The order data provides each transaction order. The trans provides transaction information and cancellation orders. To work in high-frequency transactions scenarios, the system should be very sensitive to the price and time of the data. So the data process involves two aspects: time alignment and price correction.

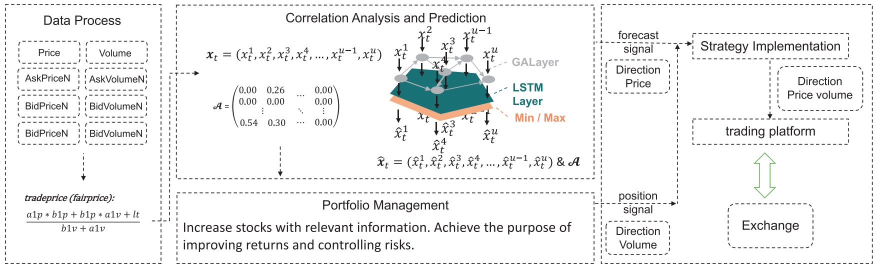


Fig. 1. Overview of the system. After data processing, we predict stock prices and analyze stock correlations. Then the strategy receives the signal of price prediction and portfolio management to generate an order and finally sends the order to the exchange through the trading platform.

Table 1
features in raw stock data.

Type	Meaning
Lastprice	The latest transaction price
Volume (Turnover)	The total volume (turnover) from today's opening to the present
BidPriceN (AskpriceN)	The current bid (ask) order price of market rank N, the bid (ask) order price rank from high to low (low to high). We use *bpN* (*apN*) to indicate it.
BidVolumeN (AskVolumeN)	The order quantity of BidPriceN (AskPriceN) price. We use *bvN* (*avN*) to indicate it.
Lastvolume (Lastturnover)	The volume (turnover) from the previous snapshot to this snapshot. We use *lv* (*lt*) to indicate it.

Tick data time alignment is an indispensable step. Because the tick time stamps given by the exchange are not accurate, we need to combine the order and trans data, and then infer the time of the data through context inference. By this way we make the market data of the same tick for each stock have the same time stamp. We mainly choose 3 seconds (tick frequency) and 1 min (the most commonly used kindle) as the research period.

In a quantitative trading system, price is always one of the most important factors. For high-frequency trading, the last price of our tick data is not accurate enough, because the data often has a huge spread. For example, in the Chinese A-share market, the unit of change is always 0.01. For a stock with a price of 3, the minimum price change is greater than three thousandths. For a product with an annualized return of 25%, it only needs to make a profit of one thousandth a day. In this case, the price is essentially discretized, and the price of any transaction cannot accurately describe the true value. Therefore, we need to define the *FairPrice*. For a trading system, the most accurate price is the actual transaction price if you intend to trade. So the *FairPrice* should be as close as possible to the real price of our transactions in the future.

To be clear, we first list a few commonly used prices and their logic, in which the variables are shown in Table 1:

Then we define

$$Dif = FairPrice - AvgPrice(T)$$

AvgPrice(T) is the average price of transactions during the future T time, we usually take $T = 1$ min. And we want to define a *FairPrice* that minimizes the absolute value of *Dif*. We find some possible solutions listed as follows:

1. ***lastprice***: the market is efficient enough, and the latest price represents the most reasonable value.
2. ***avgprice***: the average transaction price in the previous period of time, the transaction is random, the average price in a short period of time is more stable than the latest price.
3. ***midprice*** $= (ap1 + bp1)/2$: the middle price of the best bid and ask price currently offered, the current price can better reflect the expected transaction price in the future.
4. ***weightedprice*** $= (ap1 * bv1 + bp1 * av1)/(bv1 + av1)$: the real transaction price should be closer to the side that offers smaller trading volume.
5. ***tradeprice*** $= (ap1 * bv1 + bp1 * av1 + lt)/(bv1 + av1 + lv)$: the balance between the transaction price and the offer price.

In our previous trials, we find that it is more proper to take the *tradeprice* as *FairPrice*. Comprehensive consideration of liquidity, market characteristics and transaction information to reduce the impact of randomness of lastprice in the case of large spread.

After defining *FairPrice*, we can represent each stock as a sequence of *FairPrice*. In the next section, we use the multi-dimensional *FairPrice* sequences of different stocks to analysis the correlations and predict prices.

3.2. Stock correlation analysis and prediction

To analyze the correlations between different stocks and do predictions for the future with the multi-dimensional *FairPrice* sequences, we propose a novel model named GALSTM. This model can effectively extract the correlations from a graph structure, which is generated by a Multi-dimensional Hawkes Process. Then it can accurately predict the future via a novel graph processing Peephole Long Short-Term Memory. In this section, we first introduce some necessary previous models which we will use in our model. Then we will show our model in Section 3.2.4.

3.2.1. Multi-dimensional Hawkes process

The Hawkes process [2] is a kind of point process whose events can be triggered by other events. As a result, Hawkes process is also known as self-exciting process. In our approach, we use the mutually-exciting multi-dimensional Hawkes process [34], where each dimension is a Hawkes process and they have mutual excitement. The conditional intensity in the uth dimension ($u = 1, 2, \ldots, U$) is defined as follows:

$$\lambda_u^*(t) = \mu_u + \sum_{i:\, t_i < t} a_{uu_i} g(t - t_i) \tag{1}$$

where the intensity $\lambda_u^*(t)$ is the expected occurrence rate of events in the uth dimension at time t (the $*$ notation means the intensity is conditioned on the history $\mathcal{H}(t) = \{t_i | t_i < t\}$, but we omitted $\mathcal{H}(t)$ for convenience), t_i denotes the time when the ith event (maybe not in the uth dimension) happens, u_i is the dimension of the ith event, μ_u is the base intensity in the uth dimension, $a_{uu'}$ is the mutually-exciting coefficient that shows how much the events in the u'th dimension triggering those in the uth dimension, and $g(t - t_i)$ is the decay kernel that decays the impact of the ith event on time t.

Suppose there are N events happens at time $t_1, t_2, \ldots, t_N$ during time window $[0, T]$ with $0 \le t_1 < t_2 < \ldots < t_N \le T$. The log-likelihood function for the mutually-exciting multi-dimensional

Hawkes process is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) = & \sum_{i=1}^{N} \ln \lambda_{u_i}^*(t_i) - \sum_{u=1}^{U} \int_0^T \lambda_u^*(x)\mathrm{d}x \\ = & \sum_{i=1}^{N} \ln \left(\mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(t_i - t_j)\right) \\ & - \sum_{u=1}^{U} \left(\mu_u T + \sum_{i=1}^{N} a_{u u_i} G(T - t_i)\right)\end{aligned} \tag{2}$$

where parameter $\boldsymbol{\theta} = \{A, \boldsymbol{\mu}\}$, $A = (a_{uu'})$ is the matrix of the mutually-exciting coefficients, $\boldsymbol{\mu} = (\mu_u)$ is the vector of the base intensities, and $G(t) = \int_0^t g(x)\mathrm{d}x$.

3.2.2. Peephole long short-term memory

The Long Short-Term Memory (LSTM) [38] network is a kind of Recurrent Neural Network (RNN) widely used in deep learning. We will use LSTM with peephole connections [39] in our approach. Suppose the input of LSTM network is a time series $\mathbf{x}_t \in \mathbb{R}^n$ and the output is $\mathbf{h}_t \in \mathbb{R}^m$ (m, n are positive integers, $t = 1, 2, 3, \ldots$). The calculation mechanism of such an LSTM cell is as follows:

$$\begin{aligned}\mathbf{i}_t &= \sigma(W_\mathrm{i}[\mathbf{c}_{t-1} \| \mathbf{h}_{t-1} \| \mathbf{x}_t] + \mathbf{b}_\mathrm{i}) \\ \mathbf{f}_t &= \sigma(W_\mathrm{f}[\mathbf{c}_{t-1} \| \mathbf{h}_{t-1} \| \mathbf{x}_t] + \mathbf{b}_\mathrm{f}) \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(W_\mathrm{c}[\mathbf{h}_{t-1} \| \mathbf{x}_t] + \mathbf{b}_\mathrm{c}) \\ \mathbf{o}_t &= \sigma(W_\mathrm{o}[\mathbf{c}_t \| \mathbf{h}_{t-1} \| \mathbf{x}_t] + \mathbf{b}_\mathrm{o}) \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)\end{aligned} \tag{3}$$

where $\mathbf{i}_t, \mathbf{f}_t, \mathbf{c}_t, \mathbf{o}_t \in \mathbb{R}^m$, $\mathbf{c}_0, \mathbf{h}_0$ can be any vector in $\mathbb{R}^m$, $[\mathbf{c}_{t-1} \| \mathbf{h}_{t-1} \| \mathbf{x}_t] \in \mathbb{R}^{2m+n}$ and $[\mathbf{h}_{t-1} \| \mathbf{x}_t] \in \mathbb{R}^{m+n}$ are concatenated column vectors, weight matrices $W_\mathrm{i}, W_\mathrm{f}, W_\mathrm{o} \in \mathbb{R}^{m \times (2m+n)}$, $W_\mathrm{c} \in \mathbb{R}^{m \times (m+n)}$ and biases $\mathbf{b}_\mathrm{i}, \mathbf{b}_\mathrm{f}, \mathbf{b}_\mathrm{c}, \mathbf{b}_\mathrm{o} \in \mathbb{R}^m$ are parameters to update, $\sigma(\cdot)$ is an activation function, and $\odot$ is the Hadamard product. We use the following notation to denote a peephole LSTM cell (W and $\mathbf{b}$ are the abbreviation of those trainable parameters):

$$\mathbf{h}_t = \mathrm{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}; W, \mathbf{b}) \tag{4}$$

3.2.3. Graph attention layer

Considering the correlations between different stocks, the effects may be both positive or negative. To present such correlations, traditional graph neural networks are not enough. Therefore, we introduce a directed graph which can be processed by graph attention layer. The graph attention layer [40] is used to model data with directed graph structure by applying a shared attentional mechanism. Given a graph $\mathcal{G}$ with N nodes, let $\mathcal{N}_i$ ($i = 1, 2, \ldots, N$) denote the index set of the ith node's neighbors, including itself. The input of the graph attention layer is a matrix of node features $X = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)})$ ($\mathbf{x}^{(i)} \in \mathbb{R}^F$ is the features for the ith node). The output is a matrix of new node features $Y = (\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(N)})$ ($\mathbf{y}^{(i)} \in \mathbb{R}^{F'}$ is the new features for the ith node), containing the structure information of the graph. F, F' are positive integers. The propagation rule is as follows:

$$\mathbf{y}^{(i)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \mathrm{softmax}_j\left(\mathrm{LeakyReLU}\left(\mathbf{a}^\mathrm{T}\left[W\mathbf{x}^{(i)} \| W\mathbf{x}^{(j)}\right]\right)\right) W\mathbf{x}^{(j)}\right) \tag{5}$$

where $\sigma(\cdot)$ is an activation function, $\mathrm{softmax}_j(e_j) = \exp(e_j)/\sum_{k \in \mathcal{N}_j} \exp(e_k)$, negative input slope in LeakyReLU function is chosen as 0.2, $\|$ represents the concatenation operation, $\mathbf{a} \in \mathbb{R}^{2F'}$ and $W \in \mathbb{R}^{F' \times F}$ are trainable parameters. We use the following notation to denote a graph attention layer:

$$(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(N)}) = \mathrm{GAL}_\mathcal{G}(X; W, \mathbf{a}) \tag{6}$$

3.2.4. Graph attention LSTM

Next, we present our proposed GALSTM model. An overview of the GALSTM model is shown as Fig. 2. First, it detects the inherent events in each stock sequence and process them into a Multi-dimensional Hawkes Process. Then we construct a correlation graph between stock sequences using the correlation matrix which is generated by Hawkes model and limited with low-rank regulation as well as sparsity regularization. Next, we develop a graph-based attention layer to embed the input vector at each time. Finally, we use a LSTM layer combined with a min-max limitation to generate the correlations and prediction results.

Let $\{\mathbf{x}_t = (x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(U)})\}$ ($t = 1, 2, 3, \ldots$) denote the multi-dimensional *FairPrice* series obtained after data preprocessing. We now convert the time series data of stock prices to event sequences for Hawkes modeling.

Denote the volatility at time t as $\mathbf{v}_t = (v_t^{(1)}, v_t^{(2)}, \ldots, v_t^{(U)})$. For the uth dimension, we define an event as a maximal time interval with $|v_t^{(u)}| \geq \delta$ that persists more than ϵ time, where δ, ϵ are two given thresholds. We need to adjust δ and ϵ to control the number of events. The arrival time of an event is defined as the middle time of such a time interval. More specifically (ignore (u) for conciseness), let $E^+ = \{(s, t) \mid (v_{s-1}, v_{t+1} < \delta) \wedge (\forall \text{ integer } j \in [s, t], v_j \geq \delta)\}$ and $E^- = \{(s, t) \mid (v_{s-1}, v_{t+1} > -\delta) \wedge (\forall \text{ integer } j \in [s, t], v_j \leq -\delta)\}$. Then $E = \{(s, t) \in E^+ \cup E^- \mid t - s > \epsilon\}$ will be the set of events in the uth dimension. The total event set will be the union of event sets from all dimensions. For any event (s, t), the arrival time is $\tau = (s + t)\Delta/2$ by definition. Δ is the length of real time for one time step. For convenience, we sort all the events according to their arrival times, i.e., $0 \leq \tau_1 \leq \tau_2 \leq \ldots$, where τ_i is the arrival time for the ith event. Let u_i be the dimension for the ith event correspondingly.

Then we use the maximum likelihood estimation (MLE) to estimate the mutually-exciting coefficient matrix A and the base intensity vector $\boldsymbol{\mu}$. To effectively learn the Hawkes model, gradient descent algorithm can be adopted. According to Eq. (2), we have:

$$\begin{aligned}\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial a_{uu'}} &= \sum_{i: \begin{cases} u_i = u \\ 1 \leq i \leq n \end{cases}} \frac{\sum_{j:\, u_j = u' \wedge 1 \leq j \leq i-1} g(\tau_i - \tau_j)}{\mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(\tau_i - \tau_j)} - \sum_{i: \begin{cases} u_i = u' \\ 1 \leq i \leq n \end{cases}} G(\tau_n - \tau_i) \\ \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mu_u} &= \sum_{i: \begin{cases} u_i = u \\ 1 \leq i \leq n \end{cases}} \left(\mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(\tau_i - \tau_j)\right)^{-1} - \tau_n\end{aligned} \tag{7}$$

Suppose there are n events during time window $[0, \tau_n]$ in total. In addition, we add low-rank regularization $\|A\|_*$ and sparsity regularization $\|A\|_1$ to the MLE loss function [34], as not all the dimensions have impacts, where $\|\cdot\|_*$ is the nuclear norm and $\|\cdot\|_1$ is the L1 norm. Finally, we get the estimation of A, denoted as $\hat{A} = (\hat{a}_{uu'})$, a non-negative, low-rank, sparse matrix, which contains correlation information of original time series.

Now we construct a U-node correlation graph $\mathcal{A}$ according to $\hat{A}$. The uth node is the uth dimension of original time series, and the adjacent matrix $\tilde{A} = (\tilde{a}_{uu'})$ is defined as follows:

$$\tilde{a}_{uu'} = \begin{cases} 0, & \hat{a}_{uu'} = 0 \\ 1, & \hat{a}_{uu'} > 0 \end{cases} \tag{8}$$

For the prediction of time series with correlation graph $\mathcal{A}$, we propose a novel model called Graph Attention Long Short-Term Memory (GALSTM) network. GALSTM network is based on peephole LSTM in Section 3.2.2 and graph attention layer in Section 3.2.3. We embed each component $x_t^{(u)}$ in $\mathbf{x}_t \in \mathbb{R}^U$ into a feature vector $\mathscr{T}(x_t^{(u)}) \in \mathbb{R}^F$, using a reversible transformation $\mathscr{T}(\cdot)$. The matrix $X = \mathscr{T}(\mathbf{x}_t)$ is input to the networks. First, X is sent into a graph attention layer w.r.t. graph $\mathcal{A}$. Thus we get the new features $(\mathbf{y}_t^{(1)}, \mathbf{y}_t^{(2)}, \ldots, \mathbf{y}_t^{(U)})$ ($\mathbf{y}_t^{(u)} \in \mathbb{R}^{F'}$) for each node in $\mathcal{A}$, where F' is the number of features for each node. Then each $\mathbf{y}_t^{(u)}$ will be sent into a unique LSTM cell, i.e. LSTM_u, and the cell will output $\mathbf{h}_t^{(u)} \in \mathbb{R}^F$. The output of the network is $H = (\mathbf{h}_t^{(1)}, \mathbf{h}_t^{(2)}, \ldots, \mathbf{h}_t^{(U)})$. More specifically, GALSTM network can be

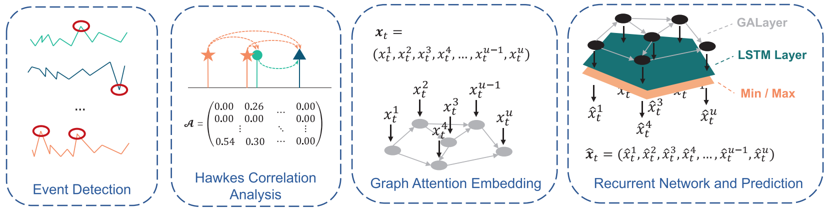


Fig. 2. An overview of the GALSTM model.

describe as follows, using Eqs. (4) and (6):

$$\begin{aligned} \left(\mathbf{y}_t^{(1)}, \mathbf{y}_t^{(2)}, \ldots, \mathbf{y}_t^{(U)}\right) &= \mathrm{GAL}_{\mathcal{A}}(X; V, \mathbf{a}) \\ \mathbf{h}_t^{(u)} &= \mathrm{LSTM}_u\left(\mathbf{y}_t^{(u)}, \mathbf{h}_{t-1}^{(u)}; W_u, \mathbf{b}_u\right) \end{aligned} \tag{9}$$

where $V, \mathbf{a}, W_u, \mathbf{b}_u$ are trainable parameters. We can just use $\hat{\mathbf{x}}_t = \mathscr{T}^{-1}(H)$ as the prediction for $\mathbf{x}_t$. However, considering the existence of high and low points in the real share market, we need to set an upper bound and a lower bound for the prediction value, which is changing every day. More specifically, assume that one day contains M time steps (i.e. $M\Delta = 24$ h) and data set starts from 0:00 of the first day. Let $\mathbf{z}_t = \mathbf{x}_{M\lfloor t/M \rfloor + 1}$ be the data at the first time step of the day that contains $\mathbf{x}_t$. Then the adjusted prediction can be expressed as follows:

$$\hat{\mathbf{x}}_t = \max\left\{\min\left\{\mathscr{T}^{-1}(H), (1+\alpha)\mathbf{z}_t\right\}, (1-\alpha)\mathbf{z}_t\right\} \tag{10}$$

where min and max are entry-wise functions, and $\alpha \geq 0$ is a given rate.

Since $\hat{\mathbf{x}}_t$ is the correlation graph where the nodes' values denote the *FairPrice* of stocks at future time t, while its edges values present the correlations between different stocks. We can learn from $\hat{\mathbf{x}}_t$ about the information of prediction and correlation for future stock markets.

3.3. Strategy implementation

To achieve a fully automatic high-frequency quantitative trading system with stable returns, our strategy should have three important features: relatively scattered trading targets, relatively short trading cycles and the trading position exposure is relatively low.

Specifically, we select hundreds of stocks and use stock index futures as a hedge. This makes our profit insensitive to the rise and fall of market. And we rely on short-term trading of a single stock to obtain profit. Because we hold a large number of stocks and the single transaction time is short, the profit or loss of a single transaction is low. This allows us to ensure that we are at a low level of risk. Once continuous losses occur we can stop all transactions in time, and the actual loss will be very small. Relying on higher winning rate, higher number of transactions and lower correlation between multiple transactions, we can get a high probability of excess return relative to the position under the premise of low risk. Coupled with the short position hedging the position return, we can obtain a stable absolute return.

In the execution of the actual trading strategy, several key factors that need to be evaluated are: stock selection, trading direction, trading price and trading volume. We mainly use the GALSTM model to update the stock position, which will be described in detail later. Model prediction is the main basis for trading direction. The order prices are controled according to some market-making strategies and high-frequency trading strategies. Comprehensive consideration of strategic positions, risks, and the entire market's trading time period, trading volume and volatility are considered to control the order volume. Finally, through our low-latency and high-frequency quantitative trading platform, we will automatically send orders and interact with the exchange about the order status information.

3.4. Portfolio management

As mentioned above, we need to hold some stock positions to obtain the intra-day trading income. In order to control the risk of holding these stocks, the best way is to copy the stocks corresponding to stock index futures. In real practice, we often do not want to fully replicate these indexes. For example, we want to find a combination with higher volatility. The common method is to control the exposure of common factors including industry, market value, etc. Here, we use the GALSTM model constructed above to adjust the positions. The short position uses the stock index futures IC contract. We calculate the matching degree d between our positions and the index in the GALSTM model. The value of d ranges from 0 to 100%. When we prepare to buy or sell a stock, we can assume that the transaction is successful and calculate the new matching degree d_{new}. We evaluate the potential drawdown caused by different matching degrees and volatility through backtesting. So we use the current volatility, d and d_{new} to calculate the current potential drawdown dd and the new potential drawdown dd_{new}. At the same time, we also estimate the previous expected return E and the new expected return E_{new} after change the stock. We will change the stock weights in our positions when the following judgment is true:

$$(E_{new} - C)/dd_{new} > (E - C)/dd$$

where C is a constant. Generally we take the value $C = 0.002$ to 0.003, and positively correlated with volatility. The specific value is determined based on backtesting.

Through the above methods, we have successfully expanded the scope of positions under risk control. We also can comprehensively judge whether to select one stock based on the different expectations of each stock and the estimated risk of the overall position.

3.5. The supporting trading platform

The underlying platform is a tool for connecting automatic trading system and exchanges. It needs to be stable and efficient. As shown in Fig. 3, our trading system includes transaction-related market data, trading interface and other modules. It also includes the research-related backtesting, historical data landing and other modules. Through the web service configuration strategy, combined with historical market information, with the help of some authentication services, the required modules are finally configured on the transaction server. We receive the market data through the broker and enter our market gateway. The market gateway combines various data sources and sends them to the strategy execution program. The strategy execution program runs on the trading server to give or cancel an order. Then send it to the broker through our trading gateway. The transaction information, such as order status, traded volume, order cancellation information, etc, are exchanged. If some risk control events occur, our system will alert us. After closing, our system will save the market data and transaction data. In addition to functionality, speed is also a very important factor for high-frequency quantitative trading. Our platform usually takes less than 2us from receiving market data to

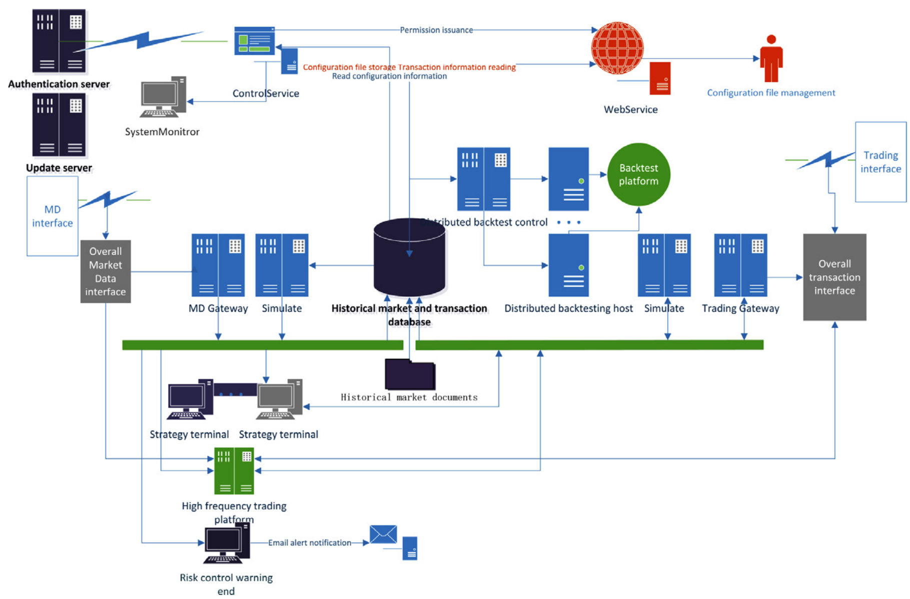


Fig. 3. High frequency trading platform.

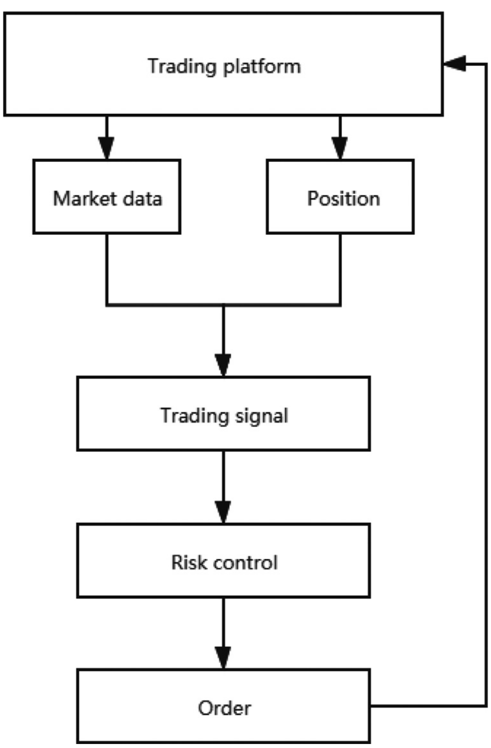


Fig. 4. Real trading flow chart. The trading platform sends market data and positions to the strategy, and the strategy generates an order based on the information, and completes the final transaction through the trading platform.

completing the order without adding additional risk control. There is no strategic time included here.

3.6. Real trading process

In actual transactions, as shown in Fig. 4, we obtain all the China A-share market data through the trading platform and combine our current positions. We may get two trading signals: changing the target positions or trading specific stocks. Changing the target positions is a long-term trading task. There are clear trading volumes, but there are no strict trading prices. When our target position changes, it usually has a difference from our current position. At this time, we will refer to the predicted prices of the specific stocks involved and make continuous adjustments until it meets the target position. Signals to trade specific stocks are the sources of income for our high-frequency strategy. Such trading signals have short action times and have clear target prices. But there are no clear trading volumes. We hope that we can trade as much as possible to pursue the maximum profits until the prices exceed the target prices. No matter what kind of trading signals will eventually produce an order, at this time the strategy will control the risk of the order. The risk control module will only modify the volume of the order. If it does not meet the risk control requirements, it will reduce the volume of the order, even to 0 in the worst case. After passing the risk control, the order will be returned to the trading platform. The trading platform sends the order to the exchange and updates the position change.

4. Experiments and case studies

In this part we show the specific experimental data and actual running results. One point that needs explanation is that the models and systems we use for real trading are constantly changing, including the iterative use of data. We usually update different modules on a daily, weekly or monthly frequency. Therefore, the following data are based on a fixed time point as an example.

4.1. Data selection and processing

We use 3-month data for almost all A-share stocks for analysis. The basic data we use is stored in the official machine room of the exchange. On this basis, we align the data and calculate *FairPrice*.

We verify our alignment and *FairPrice* adjustment methods through an auto-correlation test: since the price itself is not directly predictive, the auto-correlation of stocks' prices should be low. If there is a big difference between the lastprice and the actual price due to inaccurate time or large spreads, a certain degree of auto-correlation will occur. So we can test the results of data processing through this method. Except for a few stocks, the auto-correlation after adjustment has dropped significantly. The auto-correlation before and after adjustment is shown in Fig. 5.

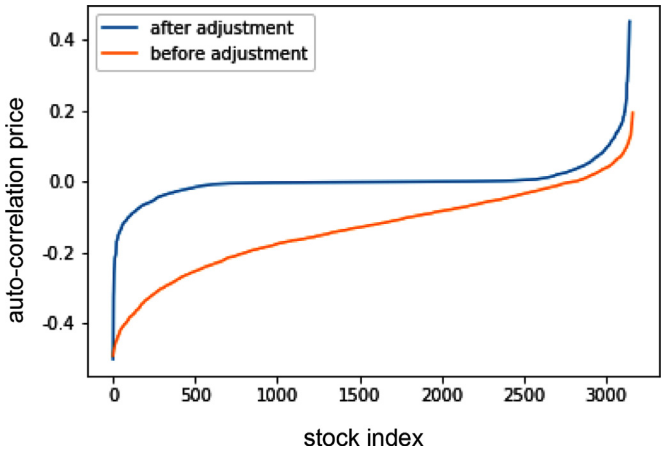


Fig. 5. Auto-correlation before and after adjustment. We calculate the auto-correlation of the 3-month tick price data for each of more than 3000 stocks, almost all of Chinas A shares. We use the lastprice and the adjusted fairprice to calculate the value of each stock separately. Finally, for easier comparison, we sort the two sets of values and draw curves.

4.2. Correlation analysis and prediction with GALSTM

With the multi-dimensional Hawkes Process result as a graph structure, we adopt GAT-encoding on normalized stock price time series to obtain the hidden states in GALSTM. We implement all the graph neural networks with the help of PyTorch, a popular and fast machine learning library. With the PyTorch Adam optimizer utilized, the MSE loss on time series prediction values could be effectively descended. We present our results both on correlation analysis and prediction to show the superb performance of the GALSTM model.

4.2.1. Correlation analysis

In this part, in order to get a graph of causality between different stock events, we train our multi-dimensional Hawkes Process model with the help of PoPPy [41] framework, which is an open source library implemented with PyTorch and CUDA acceleration. Low-rank regularization and sparse-matrix regularization are also used to improve the model.

As shown in Fig. 6, we present the correlation analysis process of GALSTM with stock index from SH600731 to SZ002428. We first use multi-dimensional Hawkes Process to extract the sub-correlation graph of these stocks as in Fig. 6a. The graph is a sparse one which shows our low-rank regularization does work. Next, we embed and update the matrix in the GALSTM model and show the final result as in Fig. 6b.

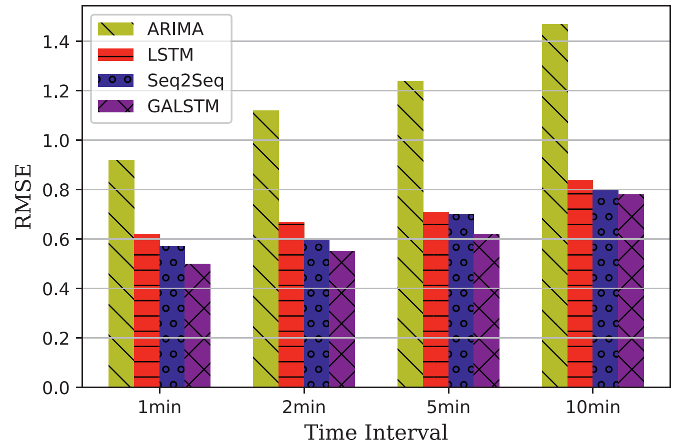


Fig. 7. Model comparison of different time intervals over test sets.

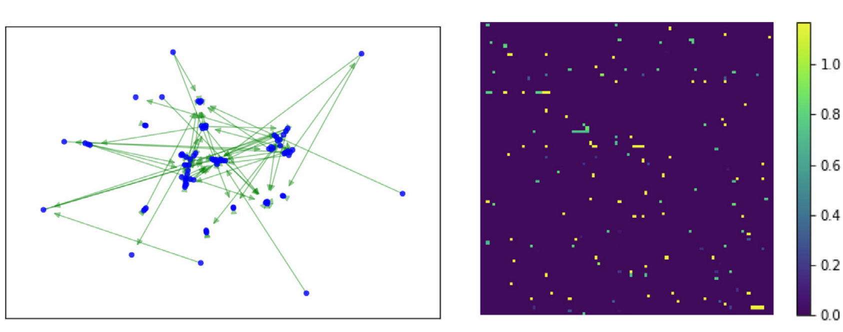
Table 2
Example causality results (from stock a to stock b) of correlation analysis for several stocks on the Shanghai and Shenzhen exchange.

Stock a	Stock b	Causality	Stock a	Stock b	Causality
SH600004	SH600057	0.2277	SZ002447	SH600851	0.0381
SH600004	SH600006	0.1559	SZ002447	SH601233	0.0163
SH600004	SH600053	0.0058	SZ002447	SH600056	0.0106
SH600006	SH600009	0.3991	SZ002449	SZ002079	0.2133
SH600006	SH600027	0.3853	SZ002449	SH600019	0.0971
SH600006	SH600959	0.3783	SZ002449	SH601058	0.0891

Inspecting our result, it first conforms to our historical experience and at the same time has made some innovations. As shown in Table 2, taking SH600004 as an example, we find that the highest correlation is SH600009, and the two stocks are related to the airport. We also find correlations between stocks in different industries. The correlations are updated in real time, so the adjustment frequency is much faster than traditional methods. On the other hand, we have not only classifications, but also quantitative values. This provides a good basis for our risk control and effectively optimizes the final capital curve.

4.2.2. Prediction

The GALSTM model adopts Graph Attention Network to embed the correlation graph into an LSTM network and update the weights of the graph constantly. The prediction for the future can be decoded from the next output graph. In this way, we can train the GALSTM model as an end-to-end frame, which is convenient and more accurate.



(a) correlation graph (b) causality matrix

Fig. 6. Part of the causality matrix uncovered by the GALSTM model between each two stocks; 100 stocks (stock index from SH600731 to SZ002428) are selected to show the causality mining result.

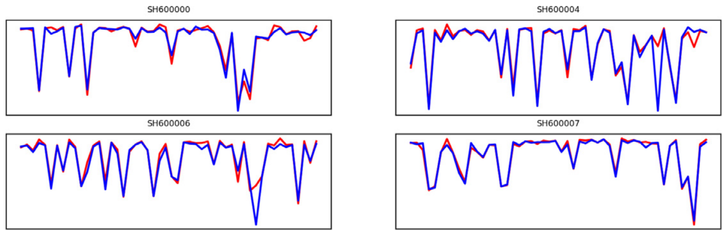


Fig. 8. Prediction of GALSTM on test set, where the red curves refer to the prediction values based on its past 50 data points and the blue curves refer to the ground truth of time series data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

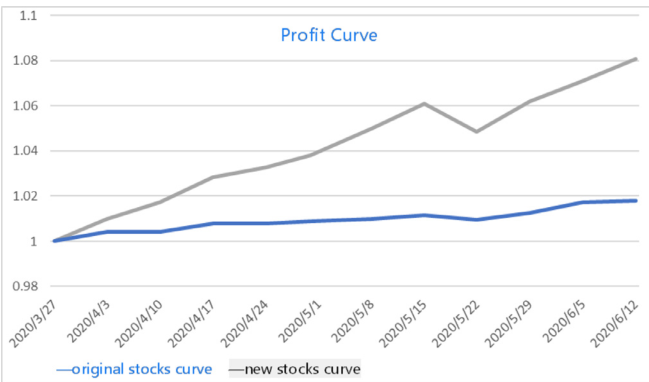


Fig. 9. Both gray and blue are our product curves, and the gray curve is the performance after using the proposed method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

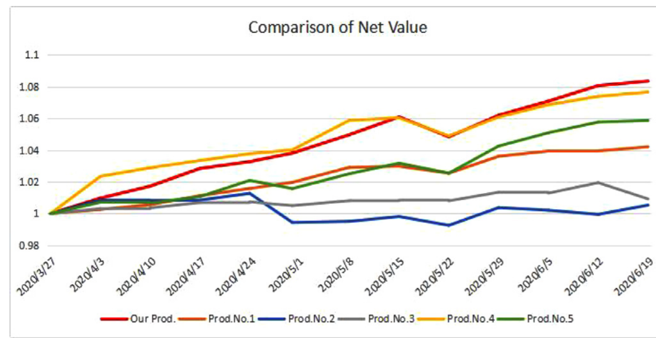


Fig. 10. The above product curves are obtained by trading real funds in the real market. Our data comes from brokers, and other data comes from third-party public platforms (access as of date 6/29/2020). The credibility is significantly higher for internal experiments.[3,4,5,6,7] Since the above is the product performance, it may include a mixed strategy in the product. Including our products, but intra-day stock trading constitutes the main source of income.

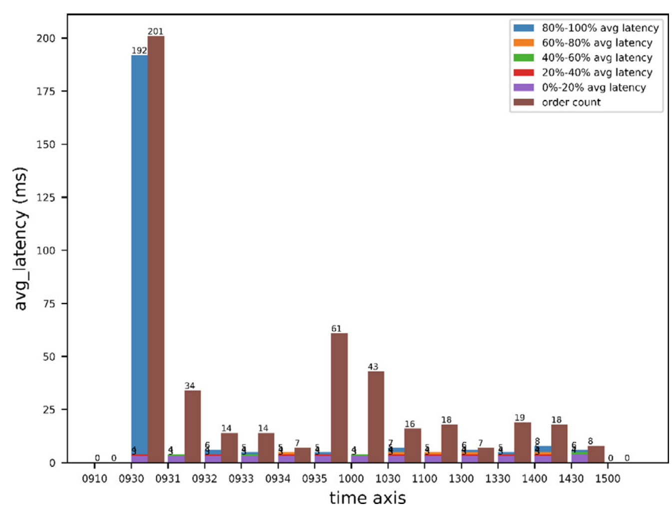


Fig. 11. Transaction latency. Considering that transaction latency mainly depends on the platform and the exchange, most trading days are similar, but the exchange will have additional delays when a small probability event such as a particularly large trading volume occurs. So we randomly select a normal trading day as a sample for analysis. We count the average delay in different time periods. Since there are more transactions when the market just opens every day, we count every minute of the first 5 min and every half hour thereafter. We separately record the number of orders corresponding to each time period and reflect the distribution of order delays through 5 colors.

GALSTM also provides accurate prediction values on our test set.

As shown in Fig. 7, during the training process of GALSTM, the MSE loss between the output and target can be effectively descended within 300 epochs of training on our training set, which consists of 3054 stocks' prices time series data in 16 days. On the basis of that, as shown in Fig. 8, the prediction of stock price data in our test set could be obtained with our GALSTM model, which shows a good accuracy.

4.3. Portfolio management and product curve

We use IC hedging in this product. We first use the stocks corresponding to the IC index as the original position. Then use the GALSTM model to estimate the correlations between stocks. Next, use stocks that are highly correlated with the original stocks and have higher expected returns to replace the original stocks in order to achieve the purpose of increasing returns under risk control. As shown in Fig. 9. We can see a significant improvement. In Figs. 9 and 10, we have normalized the starting point.

During our operation period, from March 27, 2020 to June 30, 2020, a total of 95 days. The product growth rate is 11.63% and the annualized return is 11.63%*365/95 = 44.71%, the standard deviation of daily returns is only 0.42%.

As shown in Fig. 10, we compare the products in the same period between ours and similar institutions. The comparison instances we selected are the top institutions with similar strategies in China's A-share market. It can be seen that in the same cycle, our system achieves the highest returns. Stability is also good.

In terms of performance evaluation period, considering that the strategy is intra-day trading, and the number of stocks is large, the average number of daily transactions exceeds hundreds of times, so in such a period of time, we believe it is statistically significant.

4.4. Platform robustness and speed

For high-frequency trading systems, we are most concerned about data integrity and order speed. We usually receive data through multiple brokers for integration, and then compare the data integrity of a single channel. Our trading system will analyze the market data of the channel after the market closes, and most of the loss rates can be less than one thousandth. Such errors usually have very little impact on our transactions. On the other hand, we will test our order speed. Taking the Shenzhen Stock Exchange as an example, we usually record two times: the local time T_1 when the signal is triggered, and the local time T_2 when our own order is received in the market data. We define $T_2 - T_1$ as order delay. Due to the network and other reasons, the order delay usually fluctuates, as shown in Fig. 11. The order delay is sorted into 5 sections from small to large, the average delay and the number of orders in each section are respectively counted. We can find that except the large delay in the first minute of starting the transaction, the other parts can basically be kept at about 5ms, and the worst case is also within 10 ms. The reason for the large delay in the first minute is that there are a lot of orders during this time period, and there are delays in sending market data by brokers and exchanges. Considering that the slice data of China's A-share market is updated every 3 seconds, our system performance is already sufficient to support our transactions.

5. Conclusion

In this paper, we have designed and implemented a fully automatic quantitative trading system that can achieve stable profit in a relatively long time. Our main technical advantage is that we propose a novel machine learning model named GALSTM to build a correlation graph model between stocks, which reduces the impact of the lack of hedging tools in the Chinese A-share market and improves our individual stock forecasting capabilities. We innovatively develop a dynamic stock correlation graph by a multi-dimensional Hawkes Process and fuse it into an attention-based LSTM to adaptively process and predict graph weights. Since the graph structure can focus on the salient feature of stock sequences, GALSTM can achieve a high accuracy while with a largely reduce of training time. At the same time, we design a reasonable method for dynamic adjustment of positions to maximize the effect of the above-mentioned graphical model. Cooperate with other high-frequency technology execution strategies to finally achieve the goal of high return and low risk.

In our future work, we will pay more attention to capacity issues. When our capacity becomes larger, our transactions will take longer time. This brings us two problems. First, we need longer forecasts, and we should adjust them in real time. In our GALSTM model, we currently only use the forecast value of the most recent cycle, but we can actually give the expected change in a future time period, and we will try to use the effect of longer forecasting in the future. Secondly, the financial market is highly random and may be sensitive to the initial point. Even if our expectation estimate is relatively accurate, it may still deviate significantly from the initial expectation as the market changes. At present, we only give the expected value of subsequent changes. In this case, we cannot get the optimal trading strategy. We are also interested in studying how to give the distribution of changes in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China, no. 2018AAA0100704, Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the fund from China Merchants Bank Credit Card Center.

References

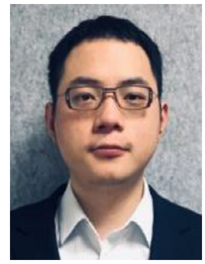
[1] X. Xiao, High frequency trading system design and process management, 2009.
[2] A. Hawkes, Point spectra of some mutually exciting point processes, J. R. Stat. Soc. Ser. B (1971) 438–443.
[3] Á.G. Zawadowski, G. Andor, J. Kertész, Short-term market reaction after extreme price changes of liquid stocks, Quant. Finance 6 (4) (2006) 283–295.
[4] Statistical arbitrage in the U.S. equities market. Marco avellaneda and Jeong-Hyun Lee. July 11, 2008.
[5] Pairs trading - a cointegration approach. Arlen David Schmidt, finance honors thesis, University of Sydney, November 2008, pages 1–130.
[6] A. d'Aspremont, Identifying small mean-reverting portfolios, Quant. Finance 11 (3) (2011) 351–364.
[7] R.D. Harris, F. Yilmaz, A momentum trading strategy based on the low frequency component of the exchange rate, J. Bank. Finance 33 (9) (2009) 1575–1585.
[8] R.J. Bianchi, M.E. Drew, J. Polichronis, A test of momentum trading strategies in foreign exchange markets: evidence from the G7, Glob. Bus. Econ. Rev. 7 (2–3) (2005) 155–179.
[9] E. Platanakis, A. Urquhart, Portfolio management with cryptocurrencies: the role of estimation risk, Econ. Lett. 177 (2019) 76–80.
[10] W.-K. Wong, J. Du, T.T.-L. Chong, Do the technical indicators reward chartists? A study on the stock markets of China, Hong Kong and Taiwan, Rev. Appl. Econ. 1 (2005) 183–205.
[11] Tulchinsky, IgorFinding alphas: a quantitative approach to building trading strategies, Wiley, NYNew York, USA, 2015.
[12] H. Wu, W. Zhang, W. Shen, J. Wang, Hybrid deep sequential modeling for social text-driven stock prediction, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, in: CIKM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1627–1630.
[13] P.C. Pendharkar, P. Cusatis, Trading financial indices with reinforcement learning agents, Expert Syst. Appl. 103 (2018) 1–13.
[14] J. Zhang, D. Maringer, Using a genetic algorithm to improve recurrent reinforcement learning for equity trading, Comput. Econ. 47 (4) (2016) 551–567.
[15] N. Ren, M. Zargham, S. Rahimi, A decision tree-based classification approach to rule extraction for security analysis, Int. J. Inf. Technol. Decis. Mak. 5 (01) (2006) 227–240.
[16] C.-F. Tsai, Y.-H. Lu, D.C. Yen, Determinants of intangible assets value: the data mining approach, Knowl. Based Syst. 31 (2012) 67–77.
[17] R. Choudhry, K. Garg, A hybrid machine learning system for stock market forecasting, World Acad. Sci. Eng. Technol. 39 (3) (2008) 315–318.
[18] V. Khatibi, E. Khatibi, A. Rasouli, A new support vector machine-genetic algorithm (SVM-GA) based method for stock market forecasting, Int. J. Phys. Sci. 6 (25) (2011) 6091–6097.
[19] A.A. Bola, A.G. Adesola, O.E. Olusayo, A.A. Adebisi, et al., Forecasting movement of the Nigerian stock exchange all share index using artificial neural and Bayesian networks, J. Finance Invest. Anal. 2 (1) (2013) 41–59.
[20] F.A. de Oliveira, C.N. Nobre, L.E. Zarate, Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index—Case study of PETR4, Petrobras, Brazil, Expert Syst. Appl. 40 (18) (2013) 7596–7606.
[21] S. Naslmosavi, A. Aghaei Chadegani, M. Mehri, Comparing the ability of Bayesian networks and adaboost for predicting financial distress of firms listed on Tehran stock exchange (TSE), Aust. J. Basic Appl. Sci. 5 (10) (2011) 629–634.
[22] S. Bogle, W. Potter, A machine learning predictive model for the Jamaica frontier market, in: Proceedings of the 2015 Intl Conference of Data Mining and Knowledge Engineering, 2015.
[23] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Syst. Appl. 42 (1) (2015) 259–268.
[24] C. Ma, Z. Liu, Z. Cao, W. Song, J. Zhang, W. Zeng, Cost-sensitive deep forest for price prediction, Pattern Recognit. 107 (2020) 107499.
[25] H. Xu, Y. Huang, Z. Duan, X. Wang, J. Feng, P. Song, Multivariate time series forecasting with transfer entropy graph, 2020, arXiv:2005.01185
[26] Y. Tao, L. Ma, W. Zhang, J. Liu, W. Liu, Q. Du, Hierarchical attention-based recurrent highway networks for time series prediction, 2018. arXiv:1806.00685
[27] J. Zhao, B. Du, L. Sun, W. Lv, Y. Liu, H. Xiong, Deep multi-task learning with relational attention for business success prediction, Pattern Recognit. 110 (2021) 107469.
[28] W. Wu, J. Yan, X. Yang, H. Zha, Discovering temporal patterns for event sequence clustering via policy mixture model, IEEE Trans. Knowl. Data Eng. (2020), doi:10.1109/TKDE.2020.2986206.
[29] J. Yan, C. Zhang, H. Zha, M. Gong, C. Sun, J. Huang, S. Chu, X. Yang, On machine learning towards predictive sales pipeline analytics, in: Twenty-ninth AAAI Conference on Artificial Intelligence, 2015.
[30] J. Lu, H. Wang, J. Zhou, Y. Chen, Z. Lai, Q. Hu, Low-rank adaptive graph embedding for unsupervised feature extraction, Pattern Recognit. (2020) 107758, doi:10.1016/j.patcog.2020.107758.
[31] F. Hu, Y. Zhu, S. Wu, W. Huang, L. Wang, T. Tan, Graphair: graph representation learning with neighborhood aggregation and interaction, Pattern Recognit. (2020) 107745 https://doi.org/10.1016/j.patcog.2020.107745.
[32] W. Wu, J. Yan, X. Yang, H. Zha, Decoupled learning for factorial marked temporal point processes, KDD, 2018.
[33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A Comprehensive Survey on Graph Neural Networks, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (2021) 4–24, doi:10.1109/TNNLS.2020.2978386..

[34] K. Zhou, H. Zha, L. Song, Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes, AISTATS, 2013.
[35] L. Li, H. Zha, Learning parametric models for social infectivity in multi-dimensional Hawkes processes, AAAI, 2014.
[36] Y. Liu, T. Yan, H. Chen, Exploiting graph regularized multi-dimensional Hawkes processes for modeling events with spatio-temporal characteristics, IJCAI, 2018.
[37] J. Shang, M. Sun, Geometric Hawkes processes with graph convolutional recurrent neural networks, AAAI, 2019.
[38] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[39] F. Gers, N.N. Schraudolph, J. Schmidhuber, Learning precise timing with LSTM recurrent networks, J. Mach. Learn. Res. 3 (2002) 115–143.
[40] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio', Y. Bengio, Graph attention networks, 2018, arXiv:1710.10903
[41] H. Xu, Poppy: a point process toolbox based on PyTorch, 2018, arXiv:1810.10122

Tao Yin Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China. Tao Yin received the bachelors degree and the MS degree in Computer Science from Shanghai Jiao Tong University, He is currently working toward the Ph.D. degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include machine learning and financial data.

Chenzhengyi Liu Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China. Chenzhengyi Liu is currently working toward the bachelor's degree in Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include machine learning and data mining.

Fangyu Ding Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China. Fangyu Ding is currently working toward the undergraduate degree in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include machine learning and temporal point process.



Ziming Feng China Merchants Bank Credit Card Center, No. 385, Ruichang Road, Shanghai, China. Ziming Feng obtained the B.E. and M.E. both in Computer Science from Shanghai Jiao Tong University in 2011 and 2014, respectively. He then joined Alibaba as an Algorithm Engineer. He is currently working as a Senior Engineer with China Merchant Bank Credit Card Center. His main research interests are computer vision and machine learning.

Bo Yuan Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China. Bo Yuan received the bachelor's degree from Peking University Medical School, Beijing, China, in 1983, and the MS degree in biochemistry, and the Ph.D. degree in molecular genetics from the University of Louisville, Louisville, Kentucky, in 1990 and 1995, respectively. He is currently a professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), Shanghai, China. Before joining SJTU, he was a Tenure-Track assistant professor with Ohio State University (OSU), Columbus, Ohio, in 2006, and served as a co-director for the OSU Program in Pharmacogenomics. At OSU, he was the founding director of OSUs genome initiative during the early 2000s, leading one of the only three independent efforts in the world (besides the Human Genome Project and the Celera Company), having assembled and deciphered the entire human and mouse genomes. His current research interests include biological networks, network evolution, stochastic process, biologically inspired computing, and bioinformatics, particularly on how these frameworks might impact the development of intelligent algorithms and systems.

Ning Zhang, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R. China. Ning Zhang received her Ph.D. from Institute of Automation, Chinese Academy of Science at 2016. Now she is doing research at department of computer science and engineering, Shanghai Jiao Tong University. The research interests include the sequence analysis and generation, multimedia understanding, reinforcement learning. She has published series of papers as first author on top journals and proceedings, including IJCAI, TNNLS, TIFS and so on.