HIST: A Graph-based Framework for Stock Trend Forecasting via Mining Concept-Oriented Shared Information

Wentao Xu[1]*, Weiqing Liu[2], Lewen Wang[2], Yingce Xia[2], Jiang Bian[2], Jian Yin[1], Tie-Yan Liu[2]
[1]Sun Yat-sen University
[2]Microsoft Research
{xuwt6@mail2,issjyin@mail}.sysu.edu.cn
{weiqing.liu,lewen.wang,yingce.xia,jiang.bian,tyliu}@microsoft.com

ABSTRACT

Stock trend forecasting, which forecasts stock prices' future trends, plays an essential role in investment. The stocks in a market can share information so that their stock prices are highly correlated. Several methods were recently proposed to mine the shared information through stock concepts (e.g., technology, Internet Retail) extracted from the Web to improve the forecasting results. However, previous work assumes the connections between stocks and concepts are stationary, and neglects the dynamic relevance between stocks and concepts, limiting the forecasting results. Moreover, existing methods overlook the invaluable shared information carried by hidden concepts, which measure stocks' commonness beyond the manually defined stock concepts. To overcome the shortcomings of previous work, we proposed a novel stock trend forecasting framework that can adequately mine the concept-oriented shared information from predefined concepts and hidden concepts. The proposed framework simultaneously utilize the stock's shared information and individual information to improve the stock trend forecasting performance. Experimental results on the real-world tasks demonstrate the efficiency of our framework on stock trend forecasting. The investment simulation shows that our framework can achieve a higher investment return than the baselines.

CCS CONCEPTS

• **Social and professional topics** → **Economic impact**; • **Applied computing** → **Forecasting**; • **Mathematics of computing** → **Graph algorithms**.

KEYWORDS

Computational Finance, Stock Trend Forecasting, Graph Mining

1 INTRODUCTION

The stock market is one of the most profitable investment channels in the real world. To pursue high yield by investing in it, stock trend forecasting has attracted increasing attention in recent years as a fundamental component of many complex investment strategies. Many of existing efforts [1, 32, 36] assume that the prices of different stocks are independent with each other and build the forecasting model merely based on information related to each stock, such as time series of historical stock price and volume (e.g., *opening price*, *closing price*, *highest price* and *trading volume*).

However, in practice, the price trends of different stocks tend to be highly correlated with each other when these stocks bear the shared concept. Such concepts are usually extracted from public company information on the Web, and can be specified by various dimensions, such as sector, industry, business, etc. For instance, most of the stocks under the high-tech concept have been sharing a similar bull trend along with the rapid development of information technology. For another example, many listed companies related to the concept of medicine have experienced drastic stock price surges after the outbreak of COVID-19 pandemic. Figure 1 illustrates some examples of stocks with their corresponding predefined concepts. Many recent studies have turned their eyes to utilizing such information in stock trend forecasting by recognizing the valuable information in stock concepts. For example, some straightforward methods [33] directly used the predefined stock concepts as input features of the linear forecasting model. In addition, some others [13, 17, 23, 27] used the same predefined concepts to form up relations between two stocks and leveraged the Graph Neural Network (GNN), whose edges are defined by conceptual relations, to build a more accurate stock trend forecasting model.

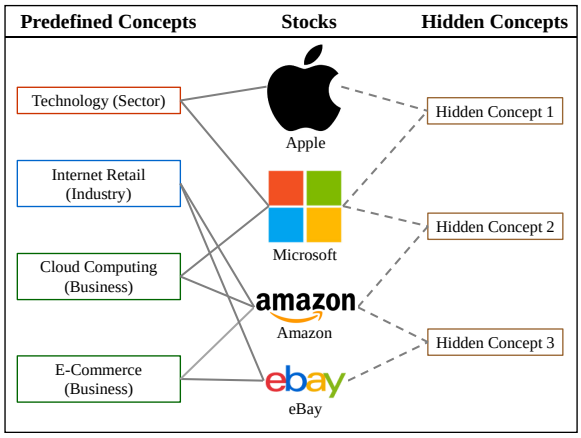


Figure 1: Some examples of stocks, predefined stock concepts and hidden stock concepts.

While these recent studies have revealed the potential of stock concepts in boosting stock trend forecasting, they are still enduring some limitations that restrain them from fully leveraging the value of stock concepts. In the first place, when building the GNN model with the predefined concepts as connections between stocks, most previous studies assume that those connections are stationary such that information propagation through them follows the same pattern. However, in the real financial market, one stock could yield a dynamic relevance degree to various concepts. For example, as shown in Figure 1, Amazon has two predefined concepts and shares the same business concept 'cloud computing' with Microsoft and

*Work done while Wentao Xu was an intern at Microsoft Research.

the other 'e-commerce' with eBay. Apparently, during the lockdown period caused by the COVID-19 pandemic, the rising stock price trend of Amazon is mainly due to its conceptual bond to surging 'e-commerce' growth[1] rather than 'cloud computing'. However, most of the existing GNN-based methods overlook differentiating the information propagation through various concepts.

Moreover, since most existing studies merely leverage the concepts predefined by human experts, they will miss some hidden concepts. Even some emerging important concepts may not be able to be promptly included in the modeling. In the meantime, some recently-listed companies may not benefit from the predefined concepts until the corresponding conceptual connections are complemented. For instance, a Personal Protective Equipment company may accidentally have shared information and similar future trend with an e-commerce company because of a sudden outbreak of a pandemic (a hidden concept representing the pandemic-related companies) but will lose this correlation after the pandemic period.

To address these limitations, we proposed a novel graph-based framework to mine the concept-oriented shared information for stock trend forecasting (HIST). The design of our HIST mainly follows the below two principles:

(1) To differentiate a stock's information propagating to related stocks with different predefined concepts, we explicitly learn the dynamic representation of various concepts by jointly modeling the relevance degrees and aggregating information from corresponding stocks. Specifically, we construct a stock-concept bipartite graph to properly extract the dynamic representations of predefined concepts and propagate them to stocks with similar representation, regardless of whether the predefined concept covers the stock.
(2) To discover the hidden concepts and the corresponding concept representations, we first introduce a doubly residual architecture [30] upon the stock-concept bipartite graph to extract the remaining information of each stock after the shared information of predefined concepts is filtered out. Subsequently, based on the stocks' remaining information, a simple yet effective graph algorithm is designed to dynamically detect hidden concepts of each time-step and construct their hidden concepts' representations.

After the information propagation via mined dynamic hidden concepts, we extract each stock's individual information by further filtering out the shared information of dynamic hidden concepts. Finally, we utilize the shared information of predefined concepts, the shared information of hidden concepts, and each stock's individual information to simultaneously forecast the stock price trend.

We evaluate our framework on real-world stock data, and the experimental results show that our framework can outperform a couple of baselines in terms of various evaluation metrics. Moreover, we simulate the stock investment using a simple but widely-used trading strategy, and the results show that our framework can achieve a higher investment return than all baselines. We also conduct additional analysis to investigate the effects of different components in our framework and visualize the mined hidden concepts to further reveal the advantages of HIST.

[1]https://www.forbes.com/sites/sergeiklebnikov/2020/07/23/5-big-numbers-that-show-amazons-explosive-growth-during-the-coronavirus-pandemic

The main contributions of this paper include:

- We proposed a new framework to mine the stocks' shared information, including the shared information of predefined concepts and hidden concepts. The shared information we mined can reflect the valuable indication of the stock's future trends with shared commonness.
- Our proposed framework can improve the stock trend forecasting performance by utilizing the dynamically shared information on the predefined and hidden concepts, and the individual information of each stock, simultaneously.
- We can discover some significant and valuable hidden concepts of stocks through our framework.
- We conducted the experimental evaluation and investment simulation on the real-world data, and the results verified our HIST framework's validity.

2 RELATED WORK

Stock trend forecasting has attracted soaring attention because it is vital in stock investment. This section will introduce two representative categories of stock trend forecasting methods: technical analysis and event-driven stock trend forecasting methods.

2.1 Technical Analysis

The technical analysis [11] predicts the stock trend based on the historical time-series of market data, such as trading price and volume. It aims to discover the trading patterns that we can leverage for future predictions. We can further divide the technical analysis methods into the **single-stock methods** and the **cross-stock methods**. The single-stock methods only use each stock's information to forecast the stock trend, and the cross-stock methods consider the cross-stock relationships between stocks when they forecast the stock price trend.

Single-stock Methods. For the single-stock methods, Autoregressive (AR) [21], and ARIMA [2] models are the most widely used model in this direction, which are both for linear and stationary time-series. However, the non-linear and non-stationary nature of stock prices limits the applicability of AR and ARIMA models. Some studies [32, 36] attempted to apply deep neural networks to catch the market trend's intricate patterns with the recent rapid development of deep learning. To further model the long-term dependency in time series, recurrent neural networks (RNN), especially Long Short-Term Memory (LSTM) network [15], had also been employed in financial predition [1, 3, 14, 34]. Specifically, [45] proposed a new State Frequency Memory (SFM) recurrent network to discover the multi-frequency trading patterns for stock price movement prediction; [20] presented a multi-task recurrent neural network with high-order Markov random fields (MRFs) to predict stock price movement direction; [12] leveraged adversarial training to simulate the stochasticity during model training. However, each stock is not isolated; the single-stock methods can not utilize the cross-stock interactions between stocks to capture more information for stock trend forecasting.

Cross-stock Methods. To mine the cross-stock shared information and improve the stock trend forecasting performance, many cross-stock methods [6, 13, 17, 27] leveraged the Graph Neural

Networks [19, 39] to capture the relationships between different stocks. The [6] and [13] utilize the graph convolutional networks (GCN) to capture the stocks' shareholder relations and industry relations, respectively. The [17] propose a hierarchical attention network for stock prediction (HATS), which uses relational data for stock market prediction. However, these existing cross-stock methods can not correctly utilize the shared information of predefined concepts because they use stationary relations to aggregate information and neglect dynamic relevance between the stocks and predefined concepts. Additionally, they also overlook the invaluable shared information of hidden concepts, limiting their stock trend forecasting performance.

2.2 Event-driven Stock Trend Forecasting

The event-driven stock trend forecasting is another category of stock trend forecasting methods, aiming to mine the event information from various sources to forecast the stock price trend. The sources of event information include the news [8, 10, 16, 28, 37, 41], social media [35, 40, 42, 43, 46], and discussion board [22, 29, 47]. These methods can discover the implicit rules governing the stock price trend from the event information. Nevertheless, the event-driven methods highly rely on the event data. The sparse and irregular event date would reduce the flexibility and performance of the event-driven stock trend forecasting models.

3 PRELIMINARIES

This section will introduce some definitions in our work and the problem of stock trend forecasting.

Definition 1. Stock concept. The predefined stock concepts are some human-defined concepts to a stock, such as the stock's sector, industry and main businesses. The hidden stock concepts are some hidden concepts that human experts do not pre-define and reflect some similar stock price trend among stocks under the same hidden concept.

Example 1. In Figure 1, there are four stocks: Apple, Microsoft, Amazon, and eBay; four predefined concepts of stocks: Technology (Sector), Internet Retail (Industry), Cloud Computing (Business), and E-Commerce (Business). There are also three hidden stock concepts among stocks. The stock Microsoft has the predefined stock concepts Technology (Sector) and Cloud Computing (Business), and the hidden stock concept Hidden concept 1 and Hidden concept 2.

Definition 2. Stock Price Trend. Many previous work [16, 41] define the stock price trend as the future change rate of the stock price. Following these settings, we define the stock price trend of stock i at date t as the stock price change rate of the next day:

$$d_i^t = \frac{Price_i^{t+1} - Price_i^t}{Price_i^t}, \tag{1}$$

where $Price_i^t$ could be specified by different values, such as the *opening price*, *closing price* and *volume weighted average price* (VWAP) [5], and we use the *closing price* in our work.

Problem 1. Stock Trend Forecasting. Given the specific stock features (e.g., the historical stock price and volume, the textual information from news and social media) of stock i at date t, the stock trend forecasting aims to forecast the stock price trend d_i^t.

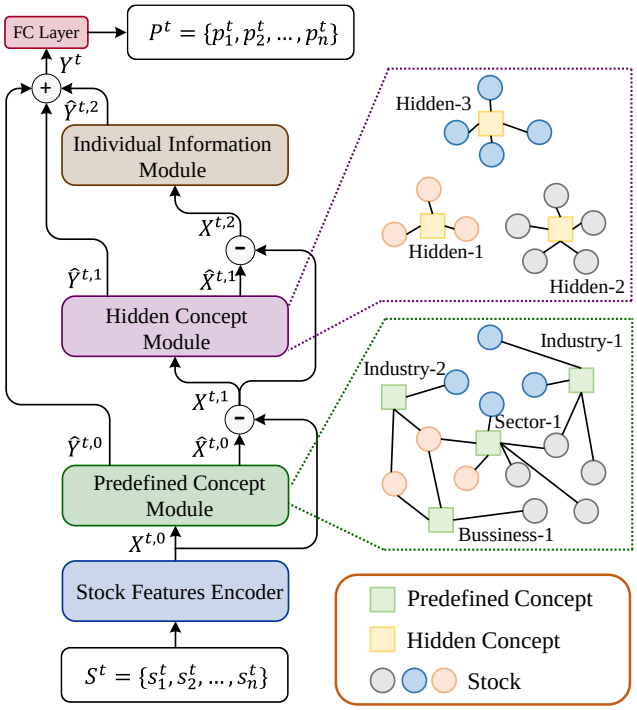


Figure 2: The overall architecture of the proposed HIST.

4 OUR HIST FRAMEWORK

4.1 Overview of Workflow

We first introduce the general workflow of HIST. The inputs of date t consists of the stock features $S^t = \{s_1^t, s_2^t, ..., s_n^t\}$ of n stocks and m predefined concepts T_j, $j \in \{1, 2, \cdots, m\}$, and our goal is to forecast the future trend of each stock. Note each stock feature s_i^t is a sequence of the historical raw feature (like opening price, closing prices, volume, etc). Figure 2 demonstrates the architecture of the HIST framework. There are three steps to use our framework:
(**Step-1**) For each stock feature s_i^t, we use a stock feature encoder to extract the temporal features of each stock. In our work, we use a 2-layer GRU network for this purpose.
(**Step-2**) Three modules will sequentially process the features obtained in Step-1 and the predefined concepts:

(1) The predefined concept module is built upon a graph neural network that extracts the shared information of stocks based on the predefined concepts. Note that the predefined concepts will be used in this step only (details in Section 4.2, Section 4.4 and 4.5).
(2) The hidden concept module, which focuses on mining the hidden shared information beyond that carried by predefined concepts (details in Section 4.3, Section 4.4 and Section 4.5).
(3) Individual information module, which processes the individual information that cannot be captured by the two kinds of shared information above (details in Section 4.6).

(**Step-3**) Finally, we feed the three types of information extracted in step 2 into a feed-forward network for prediction.

We introduce more details of the above steps:

Stock Feature Encoder. Let s_i^t denote the feature of stock i at date t, which is an l-dimensional historical stock prices and volume data. Given the stock features $S^t = \{s_1^t, s_2^t, ..., s_n^t\}$ of n stocks at date t, the stock feature encoder aims to encode the stock features to represent information for each stock in a low dimension space. Since Gated Recurrent Unit (GRU) [7] owns outstanding performance in capturing the long-term dependency, in this paper, we use a 2-layer GRU network with hidden dimension d as the stock feature encoder. Then we take the last hidden state of GRU's output as the initial embedding of the stocks at date t:

$$x_i^{t,0} = \text{GRU}\left(s_i^t\right), \tag{2}$$

where $x_i^{t,0}$ is the initial stock embedding of stock i. Then the stocks' initial embedding of all stocks at date t is the matrix $X^{t,0}$, where the i-th row of $X^{t,0}$ is $x_i^{t,0}$.

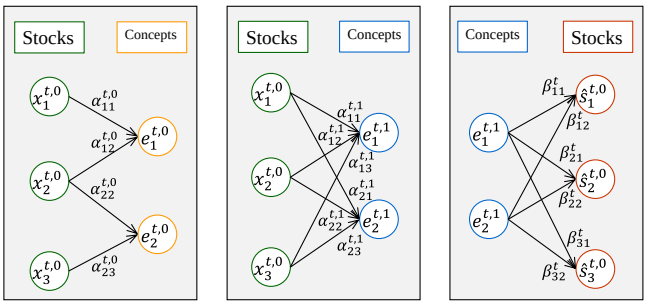
Doubly Residual Architecture. As shown in Figure 2, for Step-2 and Step-3, we follow [30] and use the doubly residual architecture. For convenience, we denote the predefined concept module, hidden concept module and individual module as module 0, 1 and 2, respectively. For each module in Step-2, given a stock i, the input of the j-th module is $x_i^{t,j}$. Each module has two outputs, one forecast output $\hat{y}_i^{t,j}$ used for the final prediction, and one backcast output $\hat{x}_i^{t,j}$ used to remove the effect of current module for the next module. Note that $x_i^{t,j} = x_i^{t,j-1} - \hat{x}_i^{t,j-1}$, $j \geq 1$, and $\hat{y}_i^t = \hat{y}_i^{t,0} + \hat{y}_i^{t,1} + \hat{y}_i^{t,2}$. For all stocks at date t, the matrix $X^{t,j} = \{x_i^{t,j}\}_{i=1}^n$ is the input of the j-th module, the matrices $\hat{X}^{t,j} = \{\hat{x}_i^{t,j}\}_{i=1}^n$ and $\hat{Y}^{t,j} = \{\hat{y}_i^t\}_{i=1}^n$ are the backcast ouput and forecast output of the j-th module, respectively.

In HIST, the predefined concept module, the hidden concept module, and the individual information module are connected in the doubly residual structure. For the backcast residual branch, the predefined concept module's backcast output $\widehat{X}^{t,0}$ and hidden concept module's backcast output $\widehat{X}^{t,1}$ are designed for removing the shared information of predefined and hidden concepts from the next module's input, making the forecasting task of downstream module easier and facilitating more fluid gradient back-propagation. For the forecast residual, the forecast outputs $\widehat{Y}^{t,0}$, $\widehat{Y}^{t,1}$ and $\widehat{Y}^{t,2}$ are finally sum up to get the stock trends forecasting.

Stock Trend Prediction. We feed the element-wise sum of the three modules' forecast output $\hat{y}_i^{t,0}$, $\hat{y}_i^{t,1}$, and $\hat{y}_i^{t,2}$ into a fully-connected layer, and output the prediction p_i^t for the stock future trend d_i^t of stock i at date t:

$$p_i^t = W_p y_i^t + b_p = W_p \left(\hat{y}_i^{t,0} + \hat{y}_i^{t,1} + \hat{y}_i^{t,2}\right) + b_p. \tag{3}$$

Following Organization. Section 4.2 to Section 4.5 will introduce details about the predefined concept module and hidden concept module. Both of these modules have three components: 1) extracting the concepts' representations from corresponding stocks; 2) aggregating the concepts' information to stocks according to the related concepts; 3) output of predefined/hidden concept module, including backcast output and forecast output. We will introduce the first component of the predefined concept module and hidden concept module in Section 4.2 and Section 4.3, respectively. The predefined concept module and the hidden concept module share the same design for the second and third components. We will present the second component in Section 4.4 and the third component in Section 4.5. Besides, the Section 4.6 and Section 4.7 describe the individual information module and training objective in details.



(a) (b) (c)

Figure 3: (a) Initializing the predefined concepts' representations (Section 4.2.1); (b) Correcting the predefined concepts' representations (Section 4.2.2); (c) Aggregating the concepts' shared information to the stocks (Section 4.4)

4.2 Extracting the Predefined Concepts' Shared Information

As mentioned in Section 1, the degree of relevance between a stock and a concept may change dynamically. To model such dynamic connections, we propose to learn the concept representations to express the rich and temporal information of predefined concepts. In this paper, we propose to obtain concept representations from the information of corresponding stocks. To be specific, we first construct a stock-concept bipartite graph with the stocks and the predefined concepts in the predefined concept module. Then we extract the predefined concepts' representation from corresponding stocks' information as following two steps.

4.2.1 Initializing the Predefined concepts' Representations. We initialize a predefined concept's representation with the weighted element-sum of stock embeddings under this concept. Since not all stocks contribute equally to a concept, referring to the calculation of the stock market index, we use the market capitalization of stock as the contribution weight from a stock to a predefined concept. More specifically, the weight from the stock i to the predefined concept T_k is:

$$\alpha_{ki}^{t,0} = \frac{c_i^t}{\sum_{j \in \mathcal{N}_k^t} c_j^t}, \tag{4}$$

where c_i^t is the market capitalization of stock i at date t, and $\mathcal{N}_k^t$ is the set of stocks that related to the concept T_k. Then we aggregate the embeddings of stocks under the same concept T_k to the concept T_k with the weight $\alpha_{ki}^{t,0}$, and $e_k^{t,0}$ is the initial representation of the predefined concept T_k:

$$e_k^{t,0} = \sum_{i \in \mathcal{N}_k^t} \alpha_{ki}^{t,0} x_i^{t,0}. \tag{5}$$

Figure 3 (a) is an example of initializing representations of the predefined concepts when stock 1 and 2 share the predefined concept T_1 and stock 2 and 3 share the predefined concept T_2.

4.2.2 Correcting the Predefined concepts' Representations. We further extract the information from the related stocks to the predefined concepts to correct concepts' representations. The design of the correcting process aims to solve two limitations of predefined concepts: 1) some of the predefined stock concepts are missing (these missing concepts nay be strongly correlated to individual stocks); 2) some concepts have little impact on the related stocks (these stocks should not be taken into considerate when process concepts' representations). We correct the predefined concepts' representations based on the similarities between all stocks and all concepts to address the two limitations. Intuitively, 1) if a stock is highly similar to a concept while the stock is not related to the concept, we assume that this concept is a missing concept for the stock. On the contrary, 2) if a stock owns a low similarity with a concept while the stock is connected to the concept, we assume that this concept is not essential.

To describe the degree of connection between stocks and concepts, we firstly compute the cosine similarity $v_{ki}^{t,0}$ between each stock embedding $x_i^{t,0}$ and each predefined concept T_k's initial representation $e_k^{t,0}$. Then we normalize the cosine similarity $v_{ki}^{t,0}$ using the softmax function and obtain the aggregated weight $\alpha_{ki}^{t,1}$ as following equation,

$$\begin{aligned} v_{ki}^{t,0} &= \text{Cosine}(x_i^{t,0}, e_k^{t,0}) = \frac{x_i^{t,0} \cdot e_k^{t,0}}{||x_i^{t,0}|| \cdot ||e_k^{t,0}||}, \\ \alpha_{ki}^{t,1} &= \frac{\exp\left(v_{ki}^{t,0}\right)}{\sum_{j \in \mathcal{S}^t} \exp\left(v_{kj}^{t,0}\right)}. \end{aligned} \tag{6}$$

Finally, we use the aggregate weights to aggregate the stocks' embeddings to correct concepts' representations. The process is demonstrated in Equation 7, where W_e and b_e are learnable parameters and LeakyReLU [26] is the activation function. Notably, the use of aggregated weight $\alpha_{ki}^{t,1}$ in Equation 7 could solve the two limitations mentioned above.

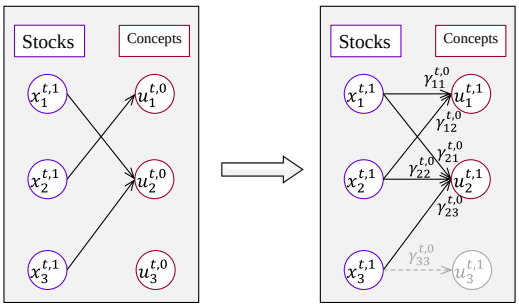
$$e_k^{t,1} = \text{LeakyReLU}\left(W_e\left(\sum_{i \in \mathcal{S}^t} \alpha_{ki}^{t,1} x_i^{t,0}\right) + b_e\right). \tag{7}$$

Figure 3 (b) is the illustration of correcting the shared information of predefined concepts.

4.3 Extracting the Hidden Concepts' Shared Information

In addition to the predefined stock concepts, some hidden stock concepts are not discovered and defined by human experts. In the hidden concept module, we utilize a simple, effective algorithm to extract the hidden concepts and corresponding representations. Similar to the predefined concept, the extracted hidden concepts' representations contain the shared information of stocks related to the same hidden concepts. As described in Section 4.1, the input of hidden concept module $x_t^{t,1}$ remove the effect of shared information $\hat{x}_i^{t,0}$ of predefined concepts from the stock embeddings $x_i^{t,0}$. The algorithm can be summarized as following steps:

(1) Initializing the hidden concepts' representations. We assume that there are n hidden concepts, which is corresponding to the n stocks. We use the stock i's embedding $x_i^{t,1}$ to initialize the corresponding hidden concept H_i's embedding $u_i^{t,0}$.



(a) (b)

Figure 4: Illustration of extracting the hidden concepts' shared information (Section 4.3).

(2) Computing the cosine similarity between all stocks and all hidden concepts with following equation:

$$\gamma_{ki}^{t,0} = \text{Cosine}(x_i^{t,1}, u_k^{t,0}) = \frac{x_i^{t,1} \cdot u_k^{t,0}}{||x_i^{t,1}|| \cdot ||u_k^{t,0}||}, \tag{8}$$

where $\gamma_{ki}^{t,0}$ is the cosine similarity between the stock i and the hidden concept H_k.

(3) Connecting stocks with hidden concepts. We connect each stock with the most similar hidden concept except for its hidden concept (initialized by this stock's embedding) and delete the hidden concepts that do not connect with any stocks. As shown in Figure 4 (a), the most similar concept of the stock embeddings $x_1^{t,1}$, $x_2^{t,1}$ and $x_3^{t,1}$ is the hidden concept H_2, H_1 and H_2, respectively, and the hidden concept H_3 will be deleted because it does not connect with any stocks.

(4) Adding connections between stocks with its own hidden concept if its concept is not deleted. Take Figure 4 (b) as an example, the stock embeddings $x_1^{t,1}$ and $x_2^{t,1}$ are connected to concepts H_1 and H_2, respectively.

(5) Obtaining the representations of hidden concepts. We utilize the cosine similarity $\gamma^{t,0}$ to aggregated information from stocks to hidden concepts, and then we get the hidden concepts' representations $u_k^{t,1}$ of hidden concept H_k:

$$u_k^{t,1} = \text{LeakyReLU}\left(W_u\left(\sum_{i \in \mathcal{M}_k^t} \gamma_{ki}^{t,0} x_i^{t,1}\right) + b_u\right), \tag{9}$$

where $\mathcal{M}_k^t$ is the set of stocks that connect to the concept H_k, W_u and b_u are learnable parameters and LeakyReLU is the activation function.

4.4 Aggregating the Concepts' Shared Information to Stocks

In both the predefined concept module and hidden concept module, we aggregate the concepts' representations to the stocks for acquiring the shared information of stocks with the same concepts. Since the shared information on different concepts has different importance for the stocks, some concepts may have higher importance

while others may have lower; we apply the attention mechanism to learn the importance of each concept for a stock.

In the predefined concept module, we first compute the cosine similarity between the stock embedding $x_i^{t,0}$ and the predefined concept T_k's representation $e_k^{t,1}$, and then normalize the similarity value $v_{ik}^{t,1}$ across all concepts in the predefined concept set $\mathcal{G}^t$ using the softmax function:

$$\begin{aligned} v_{ik}^{t,1} &= \text{Cosine}(x_i^{t,0}, e_k^{t,1}) = \frac{x_i^{t,0} \cdot e_k^{t,1}}{||x_i^{t,0}|| \cdot ||e_k^{t,1}||}, \\ \beta_{ik}^{t} &= \frac{\exp\left(v_{ik}^{t,1}\right)}{\sum_{j \in \mathcal{G}^t} \exp\left(v_{ij}^{t,1}\right)}, \end{aligned} \tag{10}$$

thus we can obtain the aggregated weight β_{ik}^t from concept representations $e_k^{t,1}$ to the stock i. We feed the aggregated information $\sum_{k \in \mathcal{G}^t} \beta_{ik}^t e_k^{t,1}$ into a fully-connected layer with LeakyReLU activation function, and obtain the shared information of the stocks correlated with the predefined concepts at date t:

$$\hat{s}_i^{t,0} = \text{LeakyReLU}\left(W_s^0 \left(\sum_{k \in \mathcal{G}^t} \beta_{ik}^t e_k^{t,1}\right) + b_s^0\right). \tag{11}$$

Figure 3 (c) illustrates an example of aggregating the concepts' shared information to the stocks. For the hidden concept module, the process of aggregating the hidden concepts' shared information is the same as the predefined concept module.

4.5 Output of Predefined/Hidden Concept Module

In predefined concept module, we feed the $\hat{s}_i^{t,0}$ into two fully-connected layers with LeakyReLU activation function to generate the backcast and forecast outputs of predefined concept module:

$$\begin{aligned} \hat{x}_i^{t,0} &= \text{LeakyReLU}\left(W_b^0 \hat{s}_i^{t,0} + b_b^0\right), \\ \hat{y}_i^{t,0} &= \text{LeakyReLU}\left(W_f^0 \hat{s}_i^{t,0} + b_f^0\right), \end{aligned} \tag{12}$$

where $\hat{x}_i^{t,0}$ and $\hat{y}_i^{t,0}$ are the backcast and forecast outputs of predefined concept module. Same as the design in the predefined concept module, the hidden concept module also has two output branches: backcast output $\hat{x}_i^{t,1}$ and the forecast output $\hat{y}_i^{t,1}$.

4.6 Individual Information Module

Beside the shared information of stocks connected with the same predefined and hidden concepts, each stock's individual information is also essential for stock trend forecasting. Therefore, we utilize the individual information module for mining the residual individual information of stocks. The input $x_i^{t,2}$ of individual information module further subtract the effect of hidden concepts' shared information $\hat{x}_i^{t,1}$. We feed the input $x_i^{t,2}$ into a fully connected layer with the LeakyReLU activation function to generate the forecast output of individual information module:

$$\hat{y}_i^{t,2} = \text{LeakyReLU}\left(W_f^2 x_i^{t,2} + b_f^2\right), \tag{13}$$

The forecast output $\hat{y}_i^{t,2}$ represents the individual information of stock i that removes the effect of stock shared information.

4.7 Training Objective Function

We leverage the Adam algorithm [18] to optimize our HIST framework by minimizing the mean squared error (MSE) loss function:

$$\mathcal{L} = \sum_{t \in \mathcal{T}} \text{MSE}\left(p^t, d^t\right) = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{S}^t} \frac{\left(p_i^t - d_i^t\right)^2}{|\mathcal{S}^t|}, \tag{14}$$

where $\mathcal{T}$ is the set of dates in the training set and the $\mathcal{S}^t$ is the set of stocks in date t; the p_i^t and d_i^t are the stock trend prediction and the ground truth stock trend of stock i at date t, respectively.

5 EXPERIMENTS

In this section, we conduct experiments for our HIST framework, aiming to answer the following research questions:

- **RQ1**: How does our model perform compared with existing stock trend forecasting methods?
- **RQ2**: What is different components' effect in our framework?
- **RQ3**: Can our model achieve a higher investment return in the investment simulation on real-world datasets?
- **RQ4**: What hidden concepts' shared information we mined?

5.1 Datasets

Stock Sets. We evaluate our HIST framework on the stocks of two popular and representative stock sets: CSI 100 and CSI 300. CSI 100 and CSI 300 comprise the largest 100 and 300 stocks in the China A-share market, respectively. The CSI 100 reflects the performances of most influential large-cap A-shares market, and the CSI 300 reflects the overall performance of China A-share market.

Stock Features. We use the stock features of Alpha360 in the open-source quantitative investment platform Qlib[2] [44]. The Alpha360 dataset contains 6 stock data on each day, which are *opening price*, *closing price*, *highest price*, *lowest price*, *volume weighted average price* (VWAP) and *trading volume*. For each stock on date t, Alpha360 looks back 60 days to construct a 360-dimensional historical stock data as a stock feature of this stock at date t. We use the features of stocks in CSI 100 and CSI 300 both from 01/01/2007 to 12/31/2020, and split them by time to obtain training set (from 01/01/2007 to 12/31/2014), validation set (from 01/01/2015 to 12/31/2016), and test set (from 01/01/2017 to 12/31/2020). We use the stock trend defined in Equation 1 as the label for each stock on each date and apply normalization on labels of the same date.

Predefined Concept Data. We collect two types of essential predefined concept data: the industry and the main business of stocks[3]. Due to the change of stocks in the stock set (the stocks in CSI 100 and CSI 300 will change half a year according to their current market capitalization) and the change of stocks' industry and businesses, the number of predefined concepts may change dynamically. The average number of predefined concepts in each day is 410 in CSI 100 and 1344 in CSI 300.

[2]https://github.com/microsoft/qlib

[3]We collect the industry and main business data from Tushare: https://tushare.pro/.

Table 1: The main results (and its standard deviation) on CSI 100 and CSI 300.

Methods	CSI 100						CSI 300					
	IC (↑)	Rank IC (↑)	Precision@N (↑)				IC (↑)	Rank IC (↑)	Precision@N (↑)			
			3	5	10	30			3	5	10	30
MLP	0.071 (4.8e-3)	0.067 (5.2e-3)	56.53 (0.91)	56.17 (0.48)	55.49 (0.30)	53.55 (0.36)	0.082 (6e-4)	0.079 (3e-4)	57.21 (0.39)	57.10 (0.33)	56.75 (0.34)	55.56 (0.14)
LSTM [15]	0.097 (2.2e-3)	0.091 (2.0e-3)	60.12 (0.52)	59.49 (0.19)	59.04 (0.15)	54.77 (0.11)	0.104 (1.5e-3)	0.098 (1.6e-3)	59.51 (0.46)	59.27 (0.34)	58.40 (0.30)	56.98 (0.11)
GRU [7]	0.103 (1.7e-3)	0.097 (1.6e-3)	59.97 (0.63)	58.99 (0.42)	58.37 (0.29)	55.09 (0.15)	0.113 (1e-3)	0.108 (8e-4)	59.95 (0.62)	59.28 (0.35)	58.59 (0.40)	57.43 (0.28)
SFM [45]	0.081 (7.0e-3)	0.074 (8.0e-3)	57.79 (0.76)	56.96 (1.04)	55.92 (0.60)	53.88 (0.47)	0.102 (3.0e-3)	0.096 (2.7e-3)	59.84 (0.91)	58.28 (0.42)	57.89 (0.45)	56.82 (0.39)
GATs [39]	0.096 (4.5e-3)	0.090 (4.4e-3)	59.17 (0.68)	58.71 (0.52)	57.48 (0.30)	54.59 (0.34)	0.111 (1.9e-3)	0.105 (1.9e-3)	60.49 (0.39)	59.96 (0.23)	59.02 (0.14)	57.41 (0.30)
ALSTM [12]	0.102 (1.8e-3)	0.097 (1.9e-3)	60.79 (0.23)	59.76 (0.42)	58.13 (0.13)	55.00 (0.12)	0.115 (1.4e-3)	0.109 (1.4e-3)	59.51 (0.20)	59.33 (0.51)	58.92 (0.29)	57.47 (0.16)
Transformer [9]	0.089 (4.7e-3)	0.090 (5.1e-3)	59.62 (1.20)	59.20 (0.84)	57.94 (0.61)	54.80 (0.33)	0.106 (3.3e-3)	0.104 (2.5e-3)	60.76 (0.35)	60.06 (0.20)	59.48 (0.16)	57.71 (0.12)
ALSTM+TRA [25]	0.107 (2.0e-3)	0.102 (1.8e-3)	60.27 (0.43)	59.09 (0.42)	57.66 (0.33)	55.16 (0.22)	0.119 (1.9e-3)	0.112 (1.7e-3)	60.45 (0.53)	59.52 (0.58)	59.16 (0.43)	58.24 (0.32)
HIST	**0.120 (1.7e-3)**	**0.115 (1.6e-3)**	**61.87 (0.47)**	**60.82 (0.43)**	**59.38 (0.24)**	**56.04 (0.19)**	**0.131 (2.2e-3)**	**0.126 (2.2e-3)**	**61.60 (0.59)**	**61.08 (0.56)**	**60.51 (0.40)**	**58.79 (0.31)**

5.2 Experimental Setting

Baselines. We compare our HIST framework with the following methods: MLP, LSTM [15], GRU [7], SFM [45], GATs [39], ALSTM [12], Transformer [9] and ALSTM+TRA [25]. The detailed introduction of these baselines are in Appendix A.1.

Evaluation Metrics. We first use two widely-used evaluation metrics: the **Information Coefficient (IC)** [25] and **Rank IC** [24]. Besides, we also use the **Precision@N**, where N is 3, 5, 10, and 30, to evaluate the precision of the top N predictions of each model. The detailed introduction of each evaluation metric is described in Appendix A.2. To eliminate the fluctuations caused by different initialization, we repeat the training and testing procedure 10 times for all results and report the average value and standard deviation.

Implementation Details. We implement our framework with the PyTorch library[4] [31], and run all experiments on a single NVIDIA Tesla V100 GPU. The hyper-parameters setting of our method and the baselines are in Appendix A.3.

5.3 Main Results (RQ1)

Table 1 shows the experimental results of HIST and other baselines on the stocks of CSI 100 and CSI 300. Our HIST framework achieves the highest IC, Rank IC, and Precision@N. Our proposed HIST framework can achieve better results than some latest stock trend forecasting methods like ALSTM, Transformer, and ALSTM+TRA; thus, our framework is more effective than existing stock trend forecasting methods. Besides, although the GATs can also capture the cross-stock connections, it only utilizes stationary cross-stock relations. Compared with GATs, our HIST can model the dynamic relevance degree between stocks and concepts; thus, it can capture the temporal and complicated cross-stock relations between stocks. Moreover, our method can further mine the shared information of hidden concepts, so our HIST outperforms existing GNN-based cross-stock technical analysis method GATs.

[4]The source code of our method and all the other baselines are available at this repository: https://github.com/Wentao-Xu/HIST.

5.4 Ablation Study (RQ2)

We apply an ablation study on our framework to study the effect of different modules in our framework. Specifically, we study the effect of initializing and correcting the predefined concept in the predefined concept module and the impact of the hidden concept module and individual information module. We study these components' effects by removing some components and observing the new experimental results. For example, when we only use the predefined concept module, we directly utilize the output of the predefined concept module to forecast the stock price trend. Table 2 shows the results of ablation study, and we have the following observations:

(1) We can find that correcting the predefined concept can improve the performance, which implies that correcting the shared information of predefined concepts, mining the missing stock concepts, and reducing the effect of concepts with less significance can improve the performance of our framework.
(2) Removing the predefined concept module or hidden concept module would reduce the performance, so the shared information of predefined and hidden concepts are both vital, and we can not ignore any one of them.
(3) Removing the individual information module would also reduce the performance, so the individual information of each stock is also crucial for stock trend forecasting.

Table 2: The results of ablation study. In this table, the *Initialize* and *Correct* are the initialization and correction of predefined concepts in predefined concept module (corresponding to Section 4.2.1 and 4.2.2), respectively. The Hidden represents the hidden concept module, and the Individual indicates the individual information module. The ✓and - indicate having or not having the component in the variants. The Precision is the average of different Precision@N values where N are 3,5,10 and 30.

Predefined		Hidden	Individual	CSI 100			CSI 300		
Initialize	*Correct*			IC (↑)	Rank IC (↑)	Precision (↑)	IC (↑)	Rank IC (↑)	Precision (↑)
✓	-	-	-	0.087	0.084	57.40	0.101	0.094	57.89
✓	✓	-	-	0.099	0.096	58.14	0.112	0.106	58.74
-	-	✓	-	0.097	0.096	58.05	0.110	0.104	58.46
✓	✓	✓	-	0.111	0.107	58.76	0.120	0.113	59.55
✓	✓	✓	✓	**0.120**	**0.115**	**59.53**	**0.131**	**0.126**	**60.50**

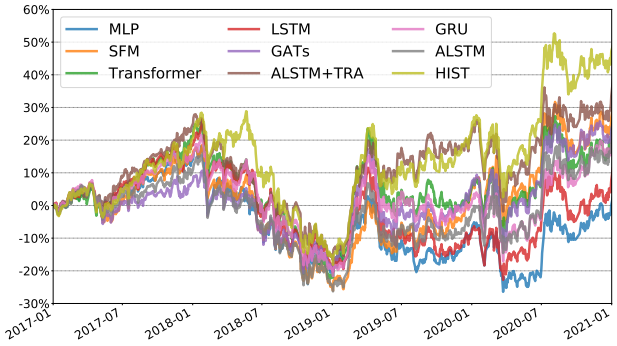


Figure 5: Cumulative Return on CSI 300 from 2017 to 2020.

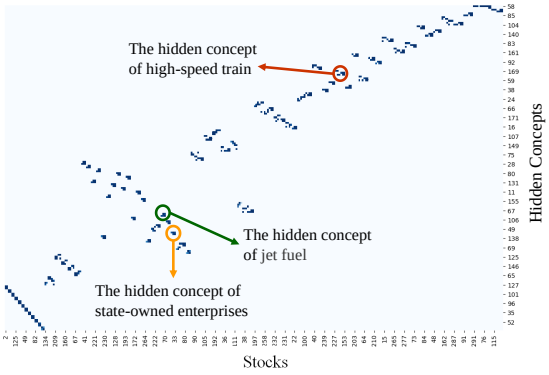


Figure 6: Visualization of the mapping between stocks and hidden concepts (a hierarchically-clustered heatmap on the weights matrix $\gamma^{t,0}$ in Section 4.3) on CSI 300 at Jan. 3, 2017. The row index indicates the stock and the column index indicates the hidden concepts, and a point (i, j) in this figure represents the stock i connects with the hidden concept H_j.

5.5 Investment Simulation (RQ3)

To further evaluate our HIST framework's effectiveness, we utilize an investment strategy to simulate the investment in the CSI 300's test set (from 01/01/2017 to 12/31/2020). To be specific, we rank the stocks on date t from high to low according to the stock trend's predictions, then select the top k stocks to evenly invest and sell the currently held stocks that not in the top k. To simulate real-world trading, we assume the initial account capital is $1e^8$, and we consider a transaction cost of 0.05% for buying shares and 0.15% for selling shares. We use the Cumulative Return to evaluate the investment simulation result:

- **Cumulative Return (CR)** is a the aggregate amount that the investment has gained or lost over time, and the cumulative return is calculated by: $\mathrm{CR} = \frac{\text{current capital} - \text{initial captial}}{\text{initial captial}}$.

To find the best selection of number k, we conduct a grid search to find the best value that maximizes the Cumulative Return on the validation set. We tune the $k \in \{10, 20, 30, 40, 50\}$, and we find that we can achieve the highest return when k is 30. Figure 5 shows the results of the Cumulative Return. Despite the stock market crash in 2018, our HIST framework still can gain a nearly 50% Cumulative Return from 2017 to 2020. Figure 5 illustrates that our HIST framework is not only outperforming some state-of-the-art methods, such as Transformer and ALSTM+TRA but also better than some graph-based models like GATs. The results of the investment simulation further confirm the efficacy of our HIST framework.

5.6 Analyzing the Hidden Concepts (RQ4)

We visualize the weights matrix $\gamma^{t,0}$ in the hidden concept module to analyze the hidden concepts we mined. Figure 6 is the hierarchically clustered heatmap on the matrix $\gamma^{t,0}$, displaying the relationships between stocks and hidden concepts. Observing the specific stocks connected with the same hidden concept can know which stocks share the same hidden information. When we enlarge Figure 6, we find the 4 stocks in the red circle include the railway construction company, the rolling stock manufacture company, and the high-speed train maintenance company, so they have the same hidden concept: 'high-speed train.' Similarly, the 3 stocks in the green circle can also have the same hidden concept 'jet fuel' because they are petrochemical companies or airlines. The predefined concepts we used do not cover these hidden concepts we mined.

6 CONCLUSION AND FUTURE WORK

We propose a graph-based framework, HIST, that can effectively mine the concept-oriented shared information among stocks and significantly improve stock trend forecasting performance. Specifically, after extracting the temporal features of stocks using GRU, we sequentially apply three modules that can extract the shared information based on the predefined concepts and the mined hidden concepts, as well as individual features that are not captured by shared features. Experimental results in the real-world stock market demonstrated the effectiveness of our proposed framework.

In the future, we plan to mine more abundant and diverse stock shared information from the Web, such as from news, social media and discussion board, and fuse different kinds of stock shared information to forecasting the stock price trend.

REFERENCES

[1] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. 2016. Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, 1–6.

[2] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. 2014. Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 106–112.

[3] Wei Bao, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one* 12, 7 (2017), e0180944.

[4] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[5] Stephen A Berkowitz, Dennis E Logue, and Eugene A Noser Jr. 1988. The total cost of transactions on the NYSE. *The Journal of Finance* 43, 1 (1988), 97–112.

[6] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating Corporation Relationship via Graph Convolutional Neural Networks for Stock Price Prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1655–1658.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[8] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z Pan, and Huajun Chen. 2019. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. In *Companion Proceedings of The 2019 World Wide Web Conference*. ACM, 678–685.

[9] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction.. In *IJCAI*. 4640–4646.

[10] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*. 2133–2142.

[11] Robert D Edwards, John Magee, and WH Charles Bassetti. 2018. *Technical analysis of stock trends*. CRC press.

[12] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training. *IJCAI* (2019).

[13] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 27.

[14] Qiyuan Gao. 2016. *Stock market forecasting using recurrent neural network*. Ph.D. Dissertation. University of Missouri–Columbia.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 261–269.

[17] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. 2019. Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999* (2019).

[18] Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[19] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[20] Chang Li, Dongjin Song, and Dacheng Tao. 2019. Multi-task Recurrent Neural Networks and Higher-order Markov Random Fields for Stock Price Movement Prediction: Multi-task RNN and Higer-order MRFs for Stock Price Classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1141–1151.

[21] Lili Li, Shan Leng, Jun Yang, and Mei Yu. 2016. Stock Market Autoregressive Dynamics: A Multinational Comparative Study with Quantile Regression. *Mathematical Problems in Engineering* 2016 (2016).

[22] Qing Li, TieJun Wang, Ping Li, Ling Liu, Qixu Gong, and Yuanzhu Chen. 2014. The effect of news and public mood on stock movements. *Information Sciences* 278 (2014), 826–840.

[23] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. [n. d.]. Modeling the stock relation with graph network for overnight stock movement prediction.

[24] Zhige Li, Derek Yang, Li Zhao, Jiang Bian, Tao Qin, and Tie-Yan Liu. 2019. Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 894–902.

[25] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. 2021. Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1017–1026.

[26] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. Citeseer, 3.

[27] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. 2019. Exploring Graph Neural Networks for Stock Market Predictions with Rolling Window Analysis. *arXiv preprint arXiv:1909.10660* (2019).

[28] Arman Khadjeh Nassirtoussi, Saeed Aghabozorgi, Teh Ying Wah, and David Chek Ling Ngo. 2015. Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Systems with Applications* 42, 1 (2015), 306–324.

[29] Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2015. Sentiment analysis on social media for stock movement prediction. *Expert Systems with Applications* 42, 24 (2015), 9603–9611.

[30] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437* (2019).

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* (2019).

[32] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. 2015. Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications* 42, 4 (2015), 2162–2172.

[33] David E Rapach, Jack K Strauss, Jun Tu, and Guofu Zhou. 2019. Industry return predictability: A machine learning approach. *The Journal of Financial Data Science* 1, 3 (2019), 9–28.

[34] Akhter Mohiuddin Rather, Arun Agarwal, and VN Sastry. 2015. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications* 42, 6 (2015), 3234–3241.

[35] Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 2013. Exploiting topic based twitter sentiment for stock prediction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 24–29.

[36] Jonathan L Ticknor. 2013. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications* 40, 14 (2013), 5501–5506.

[37] Manuel R Vargas, Beatriz SLP De Lima, and Alexandre G Evsukoff. 2017. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. IEEE, 60–65.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018). https://openreview.net/forum?id=rJXMpikCZ accepted as poster.

[40] Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. 2018. Hybrid Deep Sequential Modeling for Social Text-Driven Stock Prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. 1627–1630. https://doi.org/10.1145/3269206.3269290

[41] Wentao Xu, Weiqing Liu, Chang Xu, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. REST: Relational Event-driven Stock Trend Forecasting. *arXiv preprint arXiv:2102.07372* (2021).

[42] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1970–1979.

[43] Steve Y Yang, Sheung Yin Kevin Mo, and Anqi Liu. 2015. Twitter financial community sentiment and its predictive relationship to stock market movement. *Quantitative Finance* 15, 10 (2015), 1637–1656.

[44] Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. 2020. Qlib: An AI-oriented Quantitative Investment Platform. *arXiv preprint arXiv:2009.11189* (2020).

[45] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2141–2149.

[46] Zhenkun Zhou, Jichang Zhao, and Ke Xu. 2016. Can online emotions predict the stock market in China?. In *International conference on web information systems engineering*. Springer, 328–342.

[47] David Zimbra, Hsinchun Chen, and Robert F Lusch. 2015. Stakeholder analyses of firm-related web forums: Applications in stock return prediction. *ACM Transactions on Management Information Systems (TMIS)* 6, 1 (2015), 1–38.

A DETAILS OF EXPERIMENTAL SETTING

A.1 Baselines

We compare our proposed HIST framework with the following stock trend forecasting methods:

- **MLP**: a 3-layers multi-layer perceptron (MLP) with the number of units on each layer is 512.
- **LSTM** [15]: a Long Short-Term Memory (LSTM) network based stock trend forecasting method.
- **GRU** [7]: a Gated Recurrent Unit (GRU) network based stock trend forecasting method.
- **SFM** [45]: a RNN that decomposes the hidden states into multiple frequency components to model multi-frequency patterns.
- **GATs** [39]: a forecasting model that utilizes graph attention networks (GATs) to aggregate stock embeddings encoded by GRU on the stock graph. We use the stocks as nodes to construct a stock graph, and two stocks have a relation when they share the same predefined concept.
- **ALSTM** [12]: a variant of LSTM with a temporal attentive aggregation layer to aggregate information from all hidden states in previous timestamps.
- **Transformer** [9]: A transformer [38] based stock trend forecasting model.
- **ALSTM+TRA** [25]: an ALSTM extension that uses Temporal Routing Adaptor (TRA) to model multiple trading patterns.

A.2 Evaluation Metrics

We first use two widely used indicators in the quantitative investment domain as evaluation metrics: the **Information Coefficient (IC)** [25] and **Rank IC** [24]. The IC is the Pearson correlation coefficient [4] between the labels and predictions. The Rank IC is the Spearman's rank correlation coefficient, calculated by: $\text{Rank IC}(y^t, \hat{y}^t) = \text{corr}(rank_y^t, rank_{\hat{y}}^t)$, where $\text{corr}(\cdot)$ is the Pearson correlation coefficient. We sort the stock trend labels and predictions of all stocks on each day from high to low and acquire the ranks $rank_y^t$ and $rank_{\hat{y}}^t$ of each stock's labels and predictions on the date t. We use the average IC and Rank IC of each day to evaluate the results of stock trend forecasting.

Furthermore, for a stock trend forecasting model, the precision of its top N predictions is more vital for real-world stock investment. Therefore, we introduce another evaluation metric: **Precision@N**. The Precision@N is the proportion of top N predictions on each day with the positive label. For example, when N is 10, and the labels of 5 among these top 10 predictions are positive, then the Precision@10 is 50%. We report the results of average Precision@N on each day, and we set the N as 3, 5, 10, and 30.

A.3 Hyper-parameters Settting

The dimension l of stock features in Alpha360 is 360. and we set each training and testing batch as the stock features on the same date, so the batch size of our HIST framework and other baselines is equal to the number of stock on each day.

Besides, we tune our framework and other baselines using the grid search to select the optimal hyper-parameters based on the performance of validation set. We search the number of hidden units d of GRU in our stock features encoder, and the number of hidden units in other baselines in $\{32, 64, 128, 256, 512\}$; the number of layers of our stock feature encoder's GRU and other baselines in $\{1, 2, 3, 4\}$; the learning rate in $\{0.001, 0.0005, 0.0002, 0.0001\}$. Table 3 shows the selection of the number of units and layers in our HIST framework and other baselines, and the best learning rate in our framework and other baseline is 0.0002.

Table 3: The selection of the hyper-parameters.

Hyper-parameters	Number of Units		Number of Layers	
Dataset	CSI 100	CSI 300	CSI 100	CSI 300
MLP	512	512	3	3
LSTM	128	64	2	2
GRU	128	64	2	2
SFM	64	128	2	2
GATs	128	64	2	2
ALSTM	64	128	2	2
Transformer	32	32	3	3
ALSTM+TRA	64	128	2	2
HIST	128	128	2	2