

ST-Trader: A Spatial-Temporal Deep Neural Network for Modeling Stock Market Movement

Xiurui Hou, Kai Wang, Cheng Zhong, and Zhi Wei, *Senior Member, IEEE*

Abstract—Stocks that are fundamentally connected with each other tend to move together. Considering such common trends is believed to benefit stock movement forecasting tasks. However, such signals are not trivial to model because the connections among stocks are not physically presented and need to be estimated from volatile data. Motivated by this observation, we propose a framework that incorporates the inter-connection of firms to forecast stock prices. To effectively utilize a large set of fundamental features, we further design a novel pipeline. First, we use variational autoencoder (VAE) to reduce the dimension of stock fundamental information and then cluster stocks into a graph structure (fundamentally clustering). Second, a hybrid model of graph convolutional network and long-short term memory network (GCN-LSTM) with an adjacency graph matrix (learnt from VAE) is proposed for graph-structured stock market forecasting. Experiments on minute-level U.S. stock market data demonstrate that our model effectively captures both spatial and temporal signals and achieves superior improvement over baseline methods. The proposed model is promising for other applications in which there is a possible but hidden spatial dependency to improve time-series prediction.

Index Terms—Graph convolution network, long-short term memory network, stock market forecasting, variational autoencoder (VAE).

I. INTRODUCTION

HERE is strong evidence that stock prices of firms that interact with each other move together due to several reasons. First, exchange-traded funds (ETFs), such as S&P 500 and NASDAQ, track the prices of a basket of stocks. When people trade those funds, all the underlying stocks are traded simultaneously, which causes common fluctuations of those stock prices [1]. Second, most professional portfolio managers are specialized in a couple of strategies and these strategies often involve a similar set of stocks. For example, value investing [2] tilts to firms with high earning-to-price ratio, while momentum strategy focuses on firms with higher returns during the past year. On the one hand, any fundamental shock can affect the prices of a group of stocks

Manuscript received December 23, 2020; accepted January 19, 2021. Recommended by Associate Editor Shangce Gao. (*Corresponding author: Zhi Wei*)

Citation: X. R. Hou, K. Wang, C. Zhong, and Z. Wei, “ST-Trader: A spatial-temporal deep neural network for modeling stock market movement,” *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 5, pp. 1015–1024, May 2021.

X. R. Hou, C. Zhong, and Z. Wei are with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: xh25@njit.edu; cz352@njit.edu; zhiwei@njit.edu).

K. Wang is with the School of Business, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: kwang20@stevens.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2021.1003976

together. The ongoing COVID-19 pandemic struck the traveling industry harder than the technology sector, so we observed stocks within the traveling industry fall significantly and simultaneously while those of tech companies did not drop much. On the other hand, portfolio managers may adjust their positions due to idiosyncratic reasons, for example, the price target predicted by their own model changes. Third, some companies have cooperative relationships, like Apple and Nvidia. If one of them has good or bad news, the effect on the other one could be reflected in the stock price. To our best knowledge, however, we have not seen an algorithm effectively incorporating these hidden dependencies among firms into the stock price forecasting task.

Although the interaction among companies is not difficult to observe, it is not easy to have it cooperated in stock prices forecasting tasks due to three reasons: 1) there are too many fundamental variables to select from (usually the total is more than 1000). Extra financial expertise is needed to filter out the key variables; 2) although the key variables are selected, the interaction imposed by those variables is not trivial to model due to nonlinearity and chaos; 3) the way that interaction contributes to the final forecasting goal is the biggest obstacle to utilize the fundamental information because such static information has different frequencies and scales from time-varying price variable.

Deep learning is well deployed in grid and sequence structured data, like image recognition in autonomous driving [3] and natural language processing [4]. However, graph-formed data is more common but more complex in the real-world, like social relationships, sensor networks in smart cities, and stock connections in the financial market. Recently, there has been a surge of attention on graph representation, for example, link prediction, graph classification, and node classification. Motivated by the graph representation employed by previous studies, we propose a framework that models the hidden dependency among firms as a graph.

In the field of graph representation learning, tasks with on-hand graph information are well-studied. For example, [5] used convolutional neural networks (GCNs) to improve recommendation systems; [6] used GCNs to learn material properties from the connection of atoms directly to predict density. However, those frameworks cannot be directly applied to the stock price forecasting since it is mainly a time-series problem. Even though the stocks can be seen as nodes in a graph, edges among nodes are not trivial to define because there is no spatial locality attribute for each stock. Besides, forecasting stock price is a challenging task due to its

tumultuous nature, which prevents the transfer learning from time-series tasks with apparent seasonality or trend. Due to such irregularities, the direct application of existing graph-related models to the stock market is not appropriate.

In this work, we propose a hybrid deep learning pipeline, VAE-GCN-LSTM, to incorporate the graph structured relationship among firms into time-series forecasting tasks. The main contributions of this paper can be summarized as follows:

- 1) We design a variational autoencoder (VAE) to learn the lower-dimension latent features of companies' fundamental variables to calculate more meaningful distance among companies, which helps construct the graph network.

- 2) We develop a hybrid deep neural network of graph convolutional network and long-short term memory network (GCN-LSTM) to model both the graph structured interaction among stocks and the stock price fluctuations on the timeline.

- 3) To evaluate the contribution, and the improvement of additional features, and the proposed method, we conduct comprehensive experiments on both predicting numerical stock price and binary stock price movement on a real dataset.

We consider the largest 87 firms listed in the United States in our experiment. These firms are included in the S&P 100 index and we only incorporate 87 out of 100 firms due to the data availability. Fig. 1 shows the underlying connections among these firms. In Section III we introduce how we take the connections among firms into account in the price forecasting task, and the actual performance is presented in Section IV.

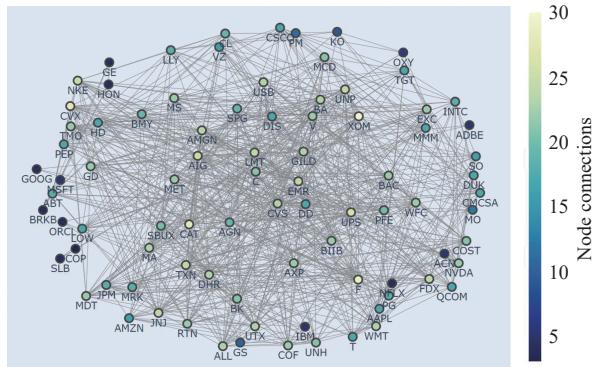


Fig. 1. Constructed network among 87 firms using VAE.

II. RELATED WORK

A. Classic Approaches in Stock Market

Many factors and firm characteristics are demonstrated to be effective in forecasting stock prices [7]. For example, financial practitioners and fund managers have been following value investing strategies after, or even prior to, the publication of "The Intelligent Investor" [8] in 1965. Since then, exploring effective factors became a hot topic and a slew of factors have been explored. Examples of these factors include the book-to-market value of a firm [9], price-to-earnings ratio [10], relative strength trading strategies that

focus on the recent stock return of a firm [11], and the profitability and investment of a firm [12]. These factors possess valuable predictive power in terms of forecasting a firm's future return.

The advent of advanced machine learning methods makes the stock return forecasting problem a more competitive task. Time-series and cross-section regressions are no longer the only toolkit we are able to utilize. For instance, [13] applied support vector machine (SVM) on forecasting NIKKEI 255 index's movement on a weekly basis. In another independent work, [14] predicted the stock price movement of Taiwan (China) companies. Reference [15] used Random Forest to forecast the future stock prices and find it outperforms both artificial neural networks (ANNs) and SVM.

B. Deep Learning Approaches in Stock Market

Deep learning has achieved exciting performance in many areas, e.g., image recognition task [16] and natural language processing task [17]. The application of deep learning also includes stock price forecasting. Both the absolute price or return (numerical prediction) and the price movement (binary classification) are popular forecasting goals for researchers. Fully connected neural networks are applied to predict future stock price in Chinese market [18] and Canadian markets [19], and to predict stock return of Japan Index [20] and S&P500 [21]. Long-short term memory (LSTM), which is designed to work on time-series tasks, is used in this forecasting problem [22], [23] since the stock price can be seen as a time series sequence. In the comparison results, LSTM has been demonstrated to outperform the fully connected neural networks.

Hybrid models, which take various sources of information to enrich the predictive power of the conventional machine learning algorithms, make the forecasting task more promising in recent years. For example, [24], [25] proposed a combination of wavelet technique and neural networks. Reference [26] introduced a hybrid model that combines autoregressive moving average models and artificial neural networks. A more recent work by [27] ensembled machine learning methods and financial technical analysis. In some related fields, like crowdfunding [28] and user intended action prediction [29], the hybridization of different types of neural networks has been applied successfully potentially due to its ability in accommodating heterogeneous input features. Reference [30] proposed a hybrid model combining long-short term memory and deep neural network (LSTM-DNN) for a stock forecasting task. This hybrid model, LSTM-DNN, first integrates static firm fundamental features into a time-series forecasting task. Refer to [31]–[34] for more hybrid models on forecasting problem. However, no matter how the neural network structure changes in the literature above, none of them takes the interactions among firms into consideration in the prediction task. It is intriguing to extract these implicit interactions and use them as input for the forecasting tasks.

C. Convolutional Neural Networks in Graph-Structure Data

Recently, applying convolutional neural networks (CNNs)

to graphs with arbitrary structures has caught people's attention. Two main directions are being explored in the literature: 1) the definition of spatial convolution is generalized in [35], [36]; 2) generalizing CNNs to 3-dimensional data as a multiplication in graph Fourier domain is discussed in [35], [37], [38] by the way of convolution theorem. Using geodesic polar coordinates, the authors define the convolution operation on meshes. Therefore, this method is suitable for manifolds and cannot be directly applied to graphs with arbitrary structures. The spatial approach proposed in [36] has more potential possibilities in generalizing CNNs to arbitrary graphs. It has three steps: first, select a target node; second, construct the neighborhood of target node; third, normalize the selected sub-graph by ordering the neighbors. The normalized sub-graphs are then fed into 1-dimension Euclidean CNN. Since there is not a natural ordering property in graphs, either temporal or spatial, it has to be imposed by a labeling procedure. The spectral framework to solve this issue is first introduced by [37] and is described in Section III-D. The main disadvantage of this method is the high computational complexity, $O(n^2)$. To overcome this problem, [38] provides strictly localized filters with a linear complexity $O(|\mathcal{E}|)$. The first order approximation of the proposed spectral filter is adopted by [39] in a semi-supervised node classification task. Thus, we also use the spectral filters introduced in [38] in our framework because of the efficiency and denote the convolution operation as $*_{\mathcal{G}}$.

Applying graph convolution operation in time-series tasks is demonstrated to be helpful in some studies. Reference [40] developed a spatio-temporal GCN model for traffic forecasting. The conventional method for traffic forecasting is to do time-series analysis for each traffic entity, like a specific highway or a city road. The more natural thought is that if two roads are close, they have high probability to experience the same volume of traffic. In their study, the GCN model can perform convolution operation with much faster training speed and fewer parameters than the traditional CNN model that is more suitable for grid structured data, e.g., images. Reference [41] also proposed a spatio-temporal GCN model for human body skeletons based action recognition and the improvement is significant. Reference [42] used GCN in dynamic texture recognition by extracting low-level features. Reference [43] combined LSTM and convolutional LSTM for capturing both time sequencing features and map sequencing features. There are some attempts to combine time-series forecasting and graph structured convolution operation, like [44] and [45] forecasting wind speed and solar radiation by enriching the time-series with wind farm distances and solar site distances. The wind farms located nearby are supposed to experience much the same wind speed and direction. Wind forecasting tasks benefit from the geographical information via graph convolutional network.

III. METHOD

In this section, we start with the problem formulation of prediction task with spatial and temporal dependencies. Then, we use VAE to reduce dimensions of fundamental feature space and generate the spatial graph structure for the selected

stocks. After that, we introduce a spatial-temporal model, GCN-LSTM, to predict the future stock prices.

A. Problem Formulation

Consider N stocks, each with M fundamental variables. The fundamentals of stock i can be represented by a vector $F_i = \{f_i^{(1)}, f_i^{(2)}, \dots, f_i^{(M)}\} \in \mathbb{R}^M$. Moreover, the fundamental matrix for all stocks is $\mathbf{F} = \{F_1, F_2, \dots, F_N\} \in \mathbb{R}^{N \times M}$. There is not a time subscript since fundamental features are not time-varying. With fundamental matrix, \mathbf{F} , we construct a graph, $\mathcal{G} = (V, E)$, where V is the set of $|V| = N$ vertices (stocks), E is the set of edges representing the spatial relationship between stocks learned from function $E = \phi(\mathbf{F})$, and $E \in \mathbb{R}^{N \times N}$.

For temporal dependency, if the look-back window is T , a time-series vector for stock i can be represented as $X_i = \{x_i^1, x_i^2, \dots, x_i^T\} \in \mathbb{R}^T$. Then, we express the data collected for all stocks in a matrix $\mathbf{X} = \{X_1, X_2, \dots, X_N\} \in \mathbb{R}^{N \times T}$.

Our goal is to forecast the price of each stock in the next time point. With both spatial and temporal features ready, we design a hybrid model of graph convolutional network and long-short term memory network (GCN-LSTM), denoted as $f_{\theta}(\cdot)$. We further denote the true values and the predicted values as $y = \{x_1^{T+1}, x_2^{T+1}, \dots, x_N^{T+1}\}$ and $\hat{y} = \{\hat{x}_1^{T+1}, \hat{x}_2^{T+1}, \dots, \hat{x}_N^{T+1}\}$, respectively. The network parameters, θ , can be estimated as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N (f_{\theta}(\mathcal{G}, X_i) - y_i)^2. \quad (1)$$

B. Variational Autoencoder

A VAE is a kind of directed probabilistic graphical model whose posterior is approximated by a neural network. In Fig. 2, we represent the directed graphical model in the area shaded with green color. The generative process starts from the bottle-neck of VAE, z , which is a latent variable. $q_{\phi}(F'|z)$ represents the data generating process that results in the reconstructed input F' . Because the marginal likelihood is intractable, the objective function of a VAE is the variational lower-bound of the marginal likelihood of data. And the marginal likelihood is the sum over the marginal likelihood of each individual fundamental variable $\log p_{\phi}(f^{(1)}, \dots, f^{(M)}) = \sum_{i=1}^M \log p_{\phi}(f^{(i)})$. The the marginal likelihood of individual fundamental variables can be written as

$$\log p_{\phi}(f^{(i)}) = D_{KL}(q_{\psi}(z|F) \| p_{\phi}(z)) + \mathcal{L}(\phi, \psi; f^{(i)}) \quad (2)$$

where $q_{\psi}(z|F)$ is the approximate posterior and $p_{\phi}(z)$ is the prior distribution of the latent variable z . The first term on the right hand side of (2) means the KL divergence between the approximate posterior $q_{\psi}(z|F)$ and the prior $p_{\phi}(z)$. The second term is the objective variational lower-bound on the marginal likelihood of feature i . Since the KL divergence term is always greater than 0, (2) can be rewritten as follows:

$$\log p_{\phi}(f^{(i)}) \geq \mathcal{L}(\phi, \psi; f^{(i)}) \quad (3)$$

$$= E_{q_{\psi}(z|f^{(i)})} [-\log q_{\psi}(z|F) + \log p_{\phi}(F|z)] \quad (4)$$

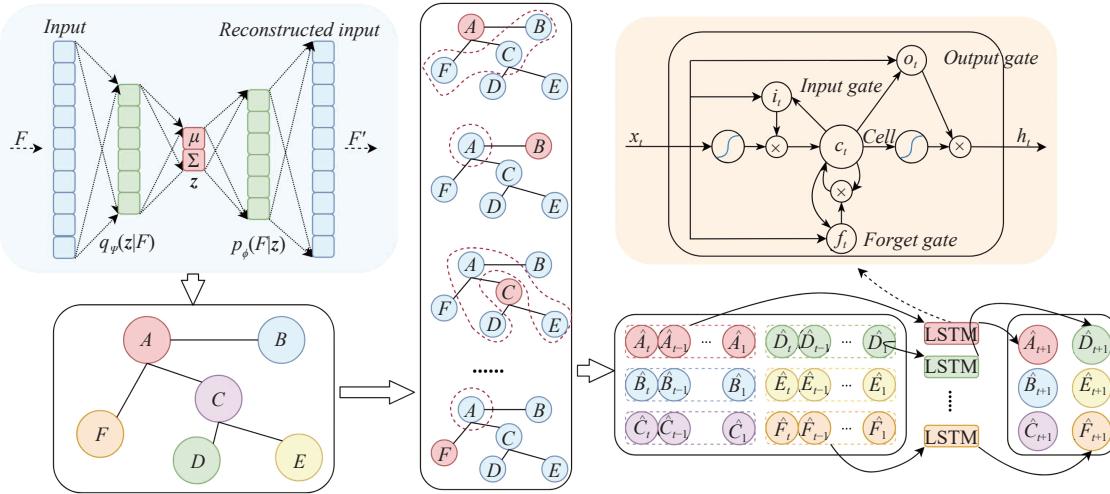


Fig. 2. Spatial-temporal modeling using GCN-LSTM framework for unspecified spatial graph structure. The area shaded in green is the VAE that reduces the dimension of the fundamental feature to learn more meaningful distance among stocks. The network below it is the constructed graph based on the learnt distance. The vertical panel to the right of VAE presents the convolution neighbors of each node. The area shaded in yellow is the network of a LSTM cell. The time-series inputs enriched with fundamental signals by convolution operation are fed into a LSTM network for final predictions.

$$= -D_{KL}(q_\psi(z|f^{(i)})||p_\phi(z)) + E_{q_\psi(z|f^{(i)})}[\log p_\phi(F|z)] \quad (5)$$

where $p_\phi(F|z)$ is the likelihood of the fundamental feature vector F given the latent variable z . The first term of (5), the KL divergence, works as a regularization term which pulls the posterior distribution to the prior distribution. The second term of (5) is the reconstruction of F through the posterior distribution $q_\psi(z|F)$ and the likelihood $p_\phi(F|z)$. The training goal is to estimate the parameters ψ and ϕ that have minimal loss between reconstruction input and original input. Ideally, the reconstructed input F' and original input F are identical.

Using the bottle-neck feature representation of VAE, we calculate the Euclidean distance between each pair of stocks, $d_{i,j}$ and define the adjacency matrix, A , as follows:

$$A_{ij} = \begin{cases} \exp(-\frac{d_{ij}^2}{\gamma^2}), & i \neq j \text{ and } \exp(-\frac{d_{ij}^2}{\gamma^2}) \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where $\gamma^2 = 0.1$ and $\epsilon = 0.5$ are thresholds to control the distribution and sparsity of adjacency matrix A . A_{ij} is the learnt distance between stock i and stock j .

C. Long-Short Term Memory (LSTM)

In Fig. 2, the construction of LSTM is represented in the area shaded in yellow color which is located in the upper-right corner. A compact form of the equations for the forward pass of a LSTM unit with a forget gate is represented below:

$$f_t = \text{sigmoid}(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \quad (7)$$

$$i_t = \text{sigmoid}(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \quad (8)$$

$$o_t = \text{sigmoid}(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \quad (9)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{c,x}x_t + W_{c,h}h_{t-1} + b_c) \quad (10)$$

$$h_t = o_t \circ \tanh(c_t) \quad (11)$$

where \circ denotes the element-wise Hadamard product; $x_t \in \mathbb{R}^T$ is the input vector at timestep t ; f_t is the forget gate activation vector; i_t and o_t are input and output gates' activation vectors, respectively; h_t and c_t represent the hidden state (output vector) and cell state. $W_{*,x} \in \mathbb{R}^{h \times T}$, $W_{*,h} \in \mathbb{R}^{h \times h}$, and $b \in \mathbb{R}^h$ are weight matrices and bias vectors.

D. Graph Convolutional Networks

We denote the convolution operation on graph \mathcal{G} as $*_{\mathcal{G}}$ and the non-parametric filter kernel as $g_\theta(\Lambda) = \text{diag}(\theta)$ with $\theta \in \mathbb{R}^N$ as a vector of Fourier coefficients. Applying filter on a graph input $x \in \mathbb{R}^{N \times T}$ can be written as

$$y = g_\theta *_{\mathcal{G}} x = g_\theta(L)x = Vg_\theta(\Lambda)V^T x \quad (12)$$

where T is the size of look-back window; V is the eigenvector matrix of the scaled Laplacian $L = I - D^{-\frac{1}{2}}AD^{\frac{1}{2}} = V\Lambda V^T$ with a diagonal matrix of its eigenvalues Λ ; The graph Fourier transform of x is denoted as $V^T x$; I is the identity matrix; $D \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$ according to [46]. Since multiplication with the eigenvector matrix V is $\mathcal{O}(N^2)$, evaluating (12) is computationally expensive. Furthermore, it is prohibitively expensive to compute the eigendecomposition of L in the first place for large graphs. To address this problem, [47] suggested an approximation for $g_\theta(\Lambda)$ via a truncated expansion in terms of Chebyshev polynomials $T_k(x)$ up to K th order

$$g_\theta(\Lambda) \approx \sum_{k=0}^K \theta'_k T_k(\tilde{\Lambda}) \quad (13)$$

with a rescaled $\tilde{\Lambda} = \frac{2}{\lambda_{\max}\Lambda} - I_N$. λ_{\max} denotes the largest eigenvalue of L and $\theta' \in \mathbb{R}^K$ is a vector of Chebyshev coefficients. The graph filtering operation can then be rewritten as follows:

$$y = g_\theta *_{\mathcal{G}} x = g_\theta(L)x = \sum_{k=0}^K \theta'_k T_k(\tilde{L})x \quad (14)$$

where $T_k(\tilde{L}) \in \mathbb{R}^{N \times N}$ represents the Chebyshev polynomial of order k evaluated at the scaled Laplacian $\tilde{L} = 2L/\lambda_{\max} - I_N$. By using the stable recurrence relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$, (14) can be approximated in $O(K|\mathcal{E}|)$ operations which is linear to the number of edges in the graph. Because the filtering operation (14) is of an order K polynomial of the Laplacian, it depends only on at most K neighbors of the target node. For details and an in-depth explanation, the reader is referred to [38].

E. ST-Trader: Spatial-Temporal Modeling Using GCN-LSTM Framework

We propose to incorporate the spatial signal into time-series prediction with graph convolutional network. The global spatial dependency learnt via VAE is represented by the adjacent matrix, A , calculated from the features from low-dimensional latent space. For notational simplicity, let $*_{\mathcal{G}} x_t$ denotes a convolution operation on x_t with filters (kernel size: $d_h \times T$) that are functions of the graph Laplacian L , as noted in (14). By replacing the original inputs with convolution-applied inputs, the equations of GCN-LSTM cell are given as follows:

$$f_t = \text{sigmoid}(W_{f,x} *_{\mathcal{G}} x_t + W_{f,h} *_{\mathcal{G}} h_{t-1} + b_f) \quad (15)$$

$$i_t = \text{sigmoid}(W_{i,x} *_{\mathcal{G}} x_t + W_{i,h} *_{\mathcal{G}} h_{t-1} + b_i) \quad (16)$$

$$o_t = \text{sigmoid}(W_{o,x} *_{\mathcal{G}} x_t + W_{o,h} *_{\mathcal{G}} h_{t-1} + b_o) \quad (17)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_{c,x} *_{\mathcal{G}} x_t + W_{c,h} *_{\mathcal{G}} h_{t-1} + b_c) \quad (18)$$

$$h_t = o_t \circ \tanh(c_t). \quad (19)$$

In our setting, $W_{\cdot,h} \in \mathbb{R}^{K \times d_h \times d_h}$ are the Chebyshev coefficients that defines the support K of the graph convolutional kernels. $W_{\cdot,x} \in \mathbb{R}^{K \times d_h \times T}$ determines the number of parameters that is independent to the number of nodes N in the graph. Parameter K determines the number of neighbors used to compute the aggregated states for any target node i so that it also determines the communication overhead in a distributed computing setting. The detailed algorithm description is shown in Algorithm 1.

Algorithm 1 Training Process for the ST-Trader

Input: $F \in \mathbb{R}^{N \times M}$. Fundamental feature matrix with M variables for N stocks;

$X \in \mathbb{R}^{N \times T}$. Time-series feature matrix with look-back window T for N stocks;

$\gamma^2 = 0.1$; $\epsilon = 0.5$; minibatch size 256; polynomial order $K = 3$; number of epochs $E = 1000$; the hidden space dimension $h = 16$; network depth $d = 3$.

Parameters: ϕ, ψ of VAE; θ of GCN-LSTM.

$\phi, \psi \leftarrow$ Initialize parameters;

repeat

$g \leftarrow \nabla_{\phi, \psi} \mathcal{L}(\phi, \psi; F)$ (Gradients of minibatch estimator of (3));

$\phi, \psi \leftarrow$ Update parameters using gradients g with learning rate $= 1e^{-5}$ and Adam optimizer [48];

until convergence of parameters (ϕ, ψ) ;

The adjacent matrix A for the edge information in $\mathcal{G} \leftarrow$ Distance of latent features derived by the VAE using parameters ϕ, ψ ;

```

 $\tilde{L} \leftarrow$  Calculate rescaled Laplacian using  $A$ ;
 $\theta \leftarrow$  Initialize Initialize parameters;
 $e \leftarrow 1$ ;
while  $e \geq E$  do
     $X' \leftarrow$  apply convolution operator  $*_{\mathcal{G}}$  on  $X$  using (14);
     $\hat{y} \leftarrow LSTM(X')$ ;
    Update  $W_x$ ,  $W_h$ , and  $b$  in  $\theta$  by gradient descent using (1) with
    learning rate  $= 1e^{-5}$  and RMSprop optimizer [49];
     $e \leftarrow e + 1$ 

```

IV. EXPERIMENT

A. Data Description

Since we consider the spatial dependency among firms, the number of our training samples is divided by the number of firms. For example, suppose we have 10 firms and 100 observations for each firm. The total number of samples is 1000 if we take firms independently but the number drops to 100 if we consider the relations among firms. To obtaining enough samples, we use minute-level stock data of 87 firms from S&P 100 composite in 2010 due to the availability of data. The number of total minute-observations is 97 890, and we split the whole dataset into batches using the sliding window. We also check the robustness of our results using five-minute-interval stock prices, which guarantees enough samples for at least one epoch training and testing. For five-minute-interval, the sample sizes of training, validation, and testing set are 16384, 2944, 2944, respectively. The used fundamental variables and stock tickers are presented in Appendix. For categorical variables such as SEC and SIC, we use their one-hot encoding as input for VAE to learn the latent feature.

B. Experimental Settings

For one-minute-level data, the testing period is one month after the training period. For example, if the testing is Feb 2010, the training data would be Jan 2010. The last testing period is Dec 2010. Thus, we have 11 testing periods and the number of samples for each month is listed in Table I.

Algorithms studying on daily data may cover a period of many years, since one year has around 250 trading dates (data points) only. However, as shown in Table I one month can have more than 7000 minute-level samples/data points. We note that the months in 2010 cover different market scenarios, such as uptrend (e.g., March), downtrend (e.g., August), and mixed (e.g., June) ones. Therefore, the minute-level data of year 2010 are representative of different market scenarios and sufficient for model validation.

To demonstrate the benefit of incorporating the spatial dependency among stocks on price forecasting, we consider the following baseline methods: 1) **LR**: For the classical linear regression model, we treat the historical time series prices as explanatory variables and the price of the next time point as the response variable; 2) **FCNN**: The fully-connected neural network which captures the non-linear relationship between time-series features; 3) **LSTM**: Long-short term memory neural network which contributes partially to the proposed method; 4) **ecldn-ST-Trader**: “ecldn” means calculating the

TABLE I
NUMBER OF SAMPLES FOR EACH MONTH

Month	Jan.	Feb.	Mar.	Apr.	May	Jun.
# of samples	7350	7410	8970	8190	7800	8680
Month	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
# of samples	8190	8580	8190	8190	7800	8580

Euclidean distance between a pair of stocks using the original fundamental variables. The spatial relationship in this setting is expected to be much noisier than using VAE; 5) **idsty_ST-Trader**: “idsty” means “industry”. The adjacent matrix for this method is derived from the industry category. $A_{ij} = 1$ means company i and company j are in the same industry category. The proposed model is denoted as **vae_ST-Trader** when compared to those baselines methods because it differs in deriving the adjacent matrix via VAE. The purpose of studying baseline methods 4) and 5) is to evaluate the ability of VAE to extract latent features from high dimension feature space. The network structure and the hyperparameters for all methods, are tuned using the validation set and the final performance results are derived on the testing set.

We apply all methods mentioned above to forecast two targets: the numerical stock price and the binary price movement indicator (the label is 1 if price goes up from last time point and 0 if price goes down from last time point). Since deep neural networks are not stable when predicting unbounded numerical results, we scale both the training set and testing set using MIN-MAX normalization (see (20)) by the maximum and minimum value of the training set.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (20)$$

C. Evaluation Metrics

For the numerical stock price prediction, we adopt widely used metrics for real-valued prediction problems [50]–[53]. They are defined as follows:

1) Mean absolute value percentage error (MAPE):

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (21)$$

where \hat{y}_i is the predicted stock price and y_i is the true stock price.

2) Median absolute value percentage error (MdAPE):

$$MdAPE = median \left(\left| \frac{\hat{y}_i - y_i}{y_i} \right| \right), i \in 1, 2, \dots, N. \quad (22)$$

MdAPE sits around the median value of the data and is thus more robust to outliers compared to MAPE.

For the binary price movement prediction, we use area under roc curve (AUC) and other following metrics, (TP for true positive, TN for true negative, FP for false positive, and FN for false negative):

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \quad (23)$$

$$Precision = \frac{TP}{TP + FP} \quad (24)$$

$$Recall = \frac{TP}{TP + FN} \quad (25)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (26)$$

D. Results

1) *Predicting Stock Price*: Table II presents the MAPE and MdAPE for all testing periods. The stock price has been demonstrated to have extensive outliers because MdAPE is usually less than MAPE. The methods enriched with spatial information achieve better prediction results than temporal-only models. The proposed model, vae_ST-Trader, outperforms baselines across the board. Moreover, LSTM is more desirable than LR and FCNN in most batches. Interestingly, idsty_ST-Trader with only industry information is more preferable than ecdn_ST-Trader, which incorporates much more fundamental information. One possibility is that simple Euclidean distance calculated from all fundamental variables brings more noise to the spatial signal because it assigns equal weight to each variable. The contribution of each variable to the final prediction is hard to be quantified by the linear model. Clearly, extracting the latent interaction (spatial distance on the latent features) among firms using VAE benefits the prediction accuracy substantially.

2) *Prediction Stock Price Movement*: Fig. 3 presents the Accuracy and AUC scores of different methods on binary movement prediction chronologically. The Efficient-market hypothesis¹ states that asset prices reflect all available information, and so there is not much space for algorithms to forecast stock prices. This hypothesis is supported by the poor accuracy in our study (Fig. 3) and in other literature. Although many investors apply value-investing strategies, which tie the market price of a stock to its underlying fundamental value, many other investors keep adopting technical analysis; they make trading decisions based on reading charts of the historical price trends. Therefore, there is still room for the methods to improve their predictive power if they do not take the influence of the technical-analysis trader on stock price into account. The proposed model is enriched by the extracted relationship among the firms so that it can better capture the trend signal compared to baseline methods. This advantage is reflected by both Accuracy and AUC scores.

The flash crash on May 6, 2010 is an example of an extreme short-term price movement in the market. Around 2:30 p.m. EST on May 6, 2010, the Dow Jones Industrial Average fell

¹ https://en.wikipedia.org/wiki/Efficient-market_hypothesis

TABLE II
MAPE (MDAPE) FOR ONE-MINUTE-INTERVAL PREDICTION

Training month	201001	201002	201003	201004	201005	201006
Testing month	201002	201003	201004	201005	201006	201007
LR	0.1579 (0.1679)	0.2056 (0.1534)	0.1684 (0.1392)	0.2313 (0.1877)	0.1672 (0.1282)	0.1641 (0.1439)
FCNN	0.1417 (0.1171)	0.227 (0.1044)	0.1418 (0.1065)	0.2314 (0.1831)	0.155 (0.1139)	0.1503 (0.1139)
LSTM	0.1077 (0.2313)	0.2145 (0.0800)	0.1512 (0.1947)	0.2283 (0.1789)	0.1058 (0.0868)	0.1398 (0.1037)
ecldn_ST-Trader	0.1206 (0.0772)	0.1018 (0.0655)	0.1605 (0.0752)	0.1857 (0.1594)	0.128 (0.0825)	0.1484 (0.1347)
idsty_ST-Trader	0.0904 (0.0699)	0.1142 (0.0714)	0.1203 (0.0619)	0.1426 (0.1319)	0.1018 (0.0741)	0.1305 (0.0899)
vae_ST-Trader	0.0789 (0.0686)	0.0827 (0.0611)	0.132 (0.0583)	0.1488 (0.1327)	0.0863 (0.0681)	0.1274 (0.0686)
Training month	201007	201008	201009	201010	201011	
Testing month	201008	201009	201010	201011	201012	
LR	0.1708 (0.1947)	0.1589 (0.1302)	0.1398 (0.1334)	0.1556 (0.1203)	0.2188 (0.1748)	
FCNN	0.1364 (0.1302)	0.1478 (0.1389)	0.1478 (0.1268)	0.1481 (0.1143)	0.1995 (0.1810)	
LSTM	0.1995 (0.1815)	0.1089 (0.0907)	0.1023 (0.0946)	0.1175 (0.1098)	0.201 (0.1369)	
ecldn_ST-Trader	0.1478 (0.1018)	0.0856 (0.0739)	0.1119 (0.0980)	0.0856 (0.0866)	0.1802 (0.1368)	
idsty_ST-Trader	0.0975 (0.0933)	0.0917 (0.0741)	0.0926 (0.0744)	0.0863 (0.0749)	0.1419 (0.1084)	
vae_ST-Trader	0.0781 (0.0816)	0.0902 (0.0728)	0.0882 (0.0606)	0.0787 (0.0627)	0.1358 (0.0780)	

Bold values indicate the best results

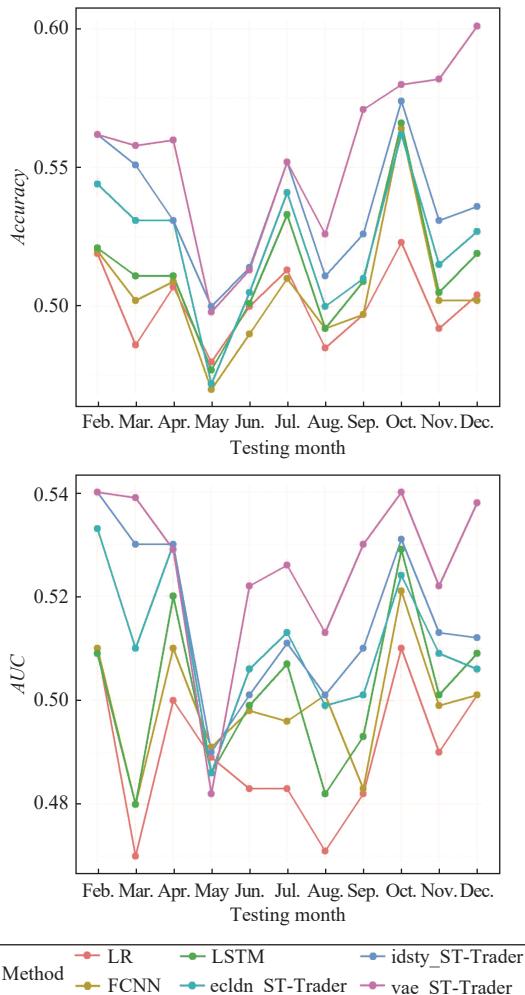


Fig. 3. ACC and AUC comparison for different methods across different testing months.

more than 1000 points in just 10 minutes². It was the biggest drop in history at that point. Despite trillions of dollars in equity being wiped out, the market recovered to its pre-crisis level by the end of the day. Analyzing the causes of such events is beyond the scope of this paper. However, it has a profound effect on the stability of our algorithm since these extreme events can hardly be thought of as shocks from fundamental information. Incorporating fundamental connections in the prediction task may lower the prediction accuracy under this particular scenario. By avoiding such events, we may expect vae_ST-Trader to perform better in predicting the longer time-interval stock price, e.g., day-to-day or week-to-week. However, we are not able to do experiments on that due to the lack of sufficient number of observations.

3) *The Number of Neighbors to Communicate With:* We compare performance of the proposed model on different granularity and different parameter K in Table III. We denote one-minute-interval and five-minute-interval price forecasting as $\Delta t = 1$ and $\Delta t = 5$. The superscript 1 and 5 highlight the best scores for $\Delta t = 1$ and $\Delta t = 5$. The results for $\Delta t = 1$ are aggregated across all testing months. There are two results worth mentioning. First, the outcomes of $\Delta t = 5$ are better than $\Delta t = 1$ for all evaluation metrics. This result is as expected because the price movement of five-minute-interval is much less noisy than one-minute-interval due to a couple of reasons: i) Rare events like the flash crash can recover so quickly that five-minute-level data can almost screen out such events; ii) The recorded stock price is bounced back and forth between the bid and ask quote and the price fluctuation in five-minute-level is less likely to be affected by the bid-ask bounce. If we have enough price observations on a longer interval, the predictive power of the proposed model can be expected to

² https://en.wikipedia.org/wiki/2010_flash_crash.

TABLE III
BINARY PRICE MOVEMENT PREDICTION WITH DIFFERENT PARAMETER K FOR ONE-MINUTE-INTERVAL AND FIVE-MINUTE-INTERVAL

		<i>ACC</i>	<i>AUC</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
$K = 3$	$\Delta t = 1$	0.567 ¹	0.523 ¹	0.469	0.619 ¹	0.534 ¹
	$\Delta t = 5$	0.577	0.556	0.474	0.411	0.440
$K = 5$	$\Delta t = 1$	0.544	0.511	0.471 ¹	0.551	0.508
	$\Delta t = 5$	0.587 ⁵	0.568 ⁵	0.492 ⁵	0.665 ⁵	0.565 ⁵
$K = 7$	$\Delta t = 1$	0.552	0.518	0.471 ¹	0.560	0.512
	$\Delta t = 5$	0.563	0.519	0.475	0.592	0.527

improve.

Second, for the communication overhead parameter K , which is the number of nodes any target node i should exchange signals with in order to derive its local states, we use different K to explore how the communication affects the performance in different granularity. One-minute-interval achieves best performance when $K = 3$ and five-minute-interval achieves best performance when $K = 5$, which means a finer granularity prefers a lighter communication to its neighbors. We give one possible explanation. For one-minute-interval, where the fluctuation of price is more random, and hence the dependency is less reliable, the communication with more neighbors brings more noise in forecasting. While along with the time interval increasing, the price trend becomes more stable and the common fluctuation is more promising so that the infusion of neighborhood signals can be more informative. Thus, the number of neighbors for supporting the center node is a key hyperparameter to tune for a specific time-series forecasting task.

V. CONCLUSION

In this paper, we propose a spatial-temporal neural network framework GCN-LSTM, to utilize the spatial dependency or the latent interaction among firms in forecasting the stock price movement. The stock market has never been treated as a graph since there is not an inborn geographical location for stock entities. However, there is strong evidence that the interactions among firms affect the stock price movement. Experimental results show that our model outperforms other state-of-the-art methods on the real-world minute-level stock price data. Fundamental features represented in a spatial structure contribute to the forecasting accuracy improvement. For future directions, we plan to investigate how the combination of fundamental variables and fiscal reports, which can be seen as a dynamic cross-section assessment of a company, contributes to predicting the stock market trend. More practical time-series applications with potential spatial dependency should be explored under the proposed modeling framework. The advanced approaches [54], [55] to the fine tuning of hyper-parameters of the proposed framework should be explored.

ACKNOWLEDGMENT

The authors would like to thank Sophia Chen for proofreading.

APPENDIX

TABLE IV
FIRM FUNDAMENTAL VARIABLES

Abbreviation	Full name
AT	Assets (Total)
INTAN	Intangible Assets (Total)
BKVLPS	BookValue Per Share
DLLT	Long Term Debt (Total)
DLC	Debt in Current Liabilities (Total)
LT	Liabilities (Total)
RE	Retained Earnings
ICAPT	Invested Capital (Total)
IB	Income Before Extraordinary Items
CHE	Cash and Short-Term Investments
PPEGT	Property, Plant and Equipment (Total)
DVT	Dividends (Total)
EBIT	Earnings Before Interest and Taxes
GP	Gross Profit (Loss)
DV	Cash Dividends (Cash Flow)
CAPX	Capital Expenditures
TXPD	Income Taxes Paid
SEC	Stock Exchange Code
SIC	Standard Industry Classification Code

TABLE V
STOCK TICKER LIST

Ticker	Ticker	Ticker
AAPL	DUK	NFLX
ABT	EMR	NKE
ACN	EXC	NVDA
ADBE	F	ORCL
AGN	FDX	OXY
AIG	GD	PEP
ALL	GE	PFE
AMGN	GILD	PG
AMZN	GOOG	PM
AXP	GS	QCOM
BA	HD	RTN
BAC	HON	SBUX

Table V (Continued)

Ticker	Ticker	Ticker
BIIB	IBM	SLB
BK	INTC	SO
BMY	JNJ	SPG
BRKB	JPM	T
C	KO	TGT
CAT	LLY	TMO
CL	LMT	TXN
CMCSA	LOW	UNH
COF	MA	UNP
COP	MCD	UPS
COST	MDT	USB
CSCO	MET	UTX
CVS	MMM	V
CVX	MO	VZ
DD	MRK	WFC
DHR	MS	WMT
DIS	MSFT	XOM

REFERENCES

- [1] I. Ben-David, F. Franzoni, and R. Moussawi, “Do ETFs increase volatility?” *J. Finance*, vol. 73, no. 6, pp. 2471–2535, Dec. 2018.
- [2] B. C. Greenwald, J. Kahn, P. D. Sonkin, and M. Van Biema, *Value Investing: From Graham to Buffett and Beyond*. New York: John Wiley & Sons, 2004.
- [3] L. Chen, X. M. Hu, W. Tian, H. Wang, D. P. Cao, and F. Y. Wang, “Parallel planning: A new motion planning framework for autonomous driving,” *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 236–246, Jan. 2019.
- [4] Z. P. Tan, J. Chen, Q. Kang, M. C. Zhou, A. Abusorrah, and K. Sedraoui, “Dynamic embedding projection-gated convolutional neural networks for text classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2021. DOI: 10.1109/TNNLS.2020.3036192.
- [5] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proc. 24th ACM SIGKDD International Conf. Knowledge Discovery & Data Mining*, London, United Kingdom, 2018, pp. 974–983.
- [6] T. Xie and J. C. Grossman, “Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties,” *Phys. Rev. Lett.*, vol. 120, no. 14, Article No. 145301, Apr. 2018.
- [7] D. Rapach and G. F. Zhou, “Forecasting stock returns,” in *Handbook of Economic Forecasting*. Amsterdam: Elsevier, 2013, vol. 2, pp. 328–383.
- [8] B. Graham, *The Intelligent Investor*. New Delhi: Prabhat Prakashan, 1965.
- [9] R. W. Banz, “The relationship between return and market value of common stocks,” *J. Finan. Econom.*, vol. 9, no. 1, pp. 3–18, Mar. 1981.
- [10] S. Basu, “Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis,” *J. Finance*, vol. 32, no. 3, pp. 663–682, Jun. 1977.
- [11] N. Jegadeesh, “Evidence of predictable behavior of security returns,” *J. Finance*, vol. 45, no. 3, pp. 881–898, Jul. 1990.
- [12] E. F. Fama and K. R. French, “A five-factor asset pricing model,” *J. Finan. Econom.*, vol. 116, no. 1, pp. 1–22, Apr. 2015.
- [13] W. Huang, Y. Nakamori, and S. Y. Wang, “Forecasting stock market movement direction with support vector machine,” *Comput. Operat. Res.*, vol. 32, no. 10, pp. 2513–2522, Oct. 2005.
- [14] Y. L. Lin, H. X. Guo, and J. L. Hu, “An SVM-based approach for stock market trend prediction,” in *Proc. Int. Joint Conf. Neural Networks*, Dallas, TX, USA, 2013, pp. 1–7.
- [15] L. Khaidem, S. Saha, and S. R. Dey, “Predicting the direction of stock market prices using random forest,” arXiv preprint arXiv: 1605.00003, 2016.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. 25th Int. Conf. Neural Information Processing Systems*, Red Hook, NY, United States, 2012, pp. 1097–1105.
- [17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv preprint arXiv: 1406.1078, 2014.
- [18] Q. Cao, K. B. Leggio, and M. J. Schniederjans, “A comparison between Fama and French’s model and artificial neural networks in predicting the Chinese stock market,” *Comput. Operat. Res.*, vol. 32, no. 10, pp. 2499–2512, Oct. 2005.
- [19] D. Olson and C. Mossman, “Neural network forecasts of Canadian stock returns using accounting ratios,” *Int. J. Forecasting*, vol. 19, no. 3, pp. 453–465, Jul.–Sept. 2003.
- [20] M. Abe and H. Nakayama, “Deep learning for forecasting stock returns in the cross-section,” in *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining*. Melbourne, VIC, Australia: Springer, 2018, pp. 273–284.
- [21] C. Krauss, X. A. Do, and N. Huck, “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500,” *Eur. J. Oper. Res.*, vol. 259, no. 2, pp. 689–702, Jun. 2017.
- [22] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.
- [23] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, “Stock market’s price movement prediction with LSTM neural networks,” in *Proc. Int. Joint Conf. Neural Networks*, Anchorage, AK, USA, 2017, pp. 1419–1426.
- [24] J. Z. Wang, J. J. Wang, Z. G. Zhang, and S. P. Guo, “Forecasting stock indices with back propagation neural network,” *Exp. Syst. Appl.*, vol. 38, no. 11, pp. 14346–14355, Oct. 2011.
- [25] T. J. Hsieh, H. F. Hsiao, and W. C. Yeh, “Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm,” *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2510–2525, Mar. 2011.
- [26] M. Khashei and M. Bijari, “An artificial neural network (p, d, q) model for timeseries forecasting,” *Exp. Syst. Appl.*, vol. 37, no. 1, pp. 479–489, Jan. 2010.
- [27] A. P. Ratto, S. Merello, L. Oneto, Y. K. Ma, L. Malandri, and E. Cambria, “Ensemble of technical analysis and machine learning for market trend prediction,” in *Proc. IEEE Symp. Series on Computational Intelligence*, Bangalore, India, 2018, pp. 2090–2096.
- [28] C. R. Cheng, F. Tan, X. R. Hou, and Z. Wei, “Success prediction on crowdfunding with multimodal deep learning,” in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, Macau, China, 2019, pp. 2158–2164.
- [29] F. Tan, Z. Wei, J. He, X. Wu, B. Peng, H. R. Liu, and Z. Y. Yan, “A blended deep learning approach for predicting user intended actions,” in *Proc. IEEE Int. Conf. Data Mining*, Singapore, 2018, pp. 487–496.
- [30] X. R. Hou, K. Wang, J. Zhang, and Z. Wei, “An enriched time-series forecasting framework for long-short portfolio strategy,” *IEEE Access*, vol. 8, pp. 31992–32002, Feb. 2020.
- [31] J. J. Wang, J. Z. Wang, Z. G. Zhang, and S. P. Guo, “Stock index forecasting based on a hybrid model,” *Omega*, vol. 40, no. 6, pp. 758–766, Dec. 2012.
- [32] Y. K. Kwon and B. R. Moon, “A hybrid neurogenetic approach for stock forecasting,” *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 851–864, May 2007.
- [33] R. de A. Araujo, A. R. L. Junior, and T. A. E. Ferreira, “A quantum-inspired intelligent hybrid method for stock market forecasting,” in *Proc. IEEE Congr. Evolutionary Computation (IEEE World Congr. Computational Intelligence)*, Hong Kong, China, 2008, pp. 1348–1355.
- [34] S. B. Wankhade, D. Surana, N. J. Mansatta, and K. Shah, “Hybrid model based on unification of technical analysis and sentiment analysis for stock price prediction,” *Int. J. Comput. Technol.*, vol. 11, no. 9,

- pp. 3025–3033, Dec. 2013.
- [35] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, “Geodesic convolutional neural networks on riemannian manifolds,” in *Proc. IEEE Int. Conf. Computer Vision Workshops*, Santiago, Chile, 2015, pp. 832–840.
- [36] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in *Proc. 33rd Int. Conf. Machine Learning*, New York, 2016, pp. 2014–2023.
- [37] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” arXiv preprint arXiv: 1312.6203, 2013.
- [38] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Red Hook, NY, United States, 2016, pp. 3844–3852.
- [39] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” arXiv preprint arXiv: 1609.02907, 2016.
- [40] B. Yu, H. T. Yin, and Z. X. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” arXiv preprint arXiv: 1709.04875, 2017.
- [41] C. L. Li, Z. Cui, W. M. Zheng, C. Y. Xu, and J. Yang, “Spatio-temporal graph convolution for skeleton based action recognition,” arXiv preprint arXiv: 1802.09834, 2018.
- [42] A. R. Rivera and O. Chae, “Spatiotemporal directional number transitional graph for dynamic texture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2146–2152, Oct. 2015.
- [43] J. T. Ke, H. Y. Zheng, H. Yang, and X. M. Chen, “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach,” *Transp. Res. C: Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.
- [44] M. Khodayar and J. H. Wang, “Spatio-temporal graph deep neural network for short-term wind speed forecasting,” *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 670–681, Apr. 2019.
- [45] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. H. Wang, and G. Y. Liu, “Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting,” *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 571–583, Apr. 2020.
- [46] F. R. K. Chung, *Spectral Graph Theory*. Providence: American Mathematical Society, 1997.
- [47] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
- [48] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv: 1412.6980, 2014.
- [49] T. Tielemans and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA*, vol. 4, no. 2, pp. 26–31, 2012.
- [50] P. F. Pai and C. S. Lin, “A hybrid ARIMA and support vector machines model in stock price forecasting,” *Omega*, vol. 33, no. 6, pp. 497–505, Dec. 2005.
- [51] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, Oct.-Dec. 2006.
- [52] S. Chopra, T. Thampy, J. Leahy, A. Caplin, and Y. LeCun, “Discovering the hidden structure of house prices with a non-parametric latent manifold model,” in *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Jose, California, USA, 2007, pp. 173–182.
- [53] D. X. Deng, C. Shahabi, U. Demiryurek, L. H. Zhu, R. Yu, and Y. Liu, “Latent space model for road networks to predict time-varying traffic,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, California, USA, 2016, pp. 1525–1534.
- [54] S. C. Gao, M. C. Zhou, Y. R. Wang, J. J. Cheng, H. Yachi, and J. H. Wang, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.
- [55] J. J. Wang and T. Kumbasar, “Parameter optimization of interval type-2 fuzzy neural networks based on PSO and BBBC methods,” *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 247–257, Jan. 2019.



Xiurui Hou received the B.E. degree in the internet of things engineering from Central South University in 2015, and the Ph.D. degree in computer science from the New Jersey Institute of Technology, USA, in 2020. Her research interests include data mining, deep learning, and computational finance.



Kai Wang received the B.E. degree in the internet of things engineering from Central South University in 2015, and the M.S. degree in financial engineering from the Stevens Institute of Technology, USA, in 2017, where he is currently pursuing the Ph.D. degree in finance. His research interests include asset pricing, market microstructure, and machine learning.



Cheng Zhong is a graduate student in the Department of Computer Science at the New Jersey Institute of Technology, USA. His research interests include data mining and deep learning.



Zhi Wei (SM’17) received the B.S. degree from Wuhan University, and the Ph.D. degree from the University of Pennsylvania, USA, in 2008. He is currently a Professor of computer science and statistics (Joint appointment) with the New Jersey Institute of Technology. He has authored or coauthored more than 150 publications with 10 000 citations and H-index of 42. His research interests include statistical modeling, machine learning, and big data analytics. He has served as a PC Member for the IEEE ICDM, ACM SIGKDD, the IEEE BigData, AAAI, and CIKM. He is an Editorial Board Member of the *IEEE Internet of Things Journal*, *BMC Genomics*, *BMC Bioinformatics*, *PLOS ONE*, and the *IEEE Transactions on Computational Social Systems*.