Using Kernel Method to Include Firm Correlation for Stock Price Prediction

Hang Xu

Shanghai University of Finance and Economics, Shanghai, China

Correspondence should be addressed to Hang Xu; xxuhang@163.sufe.edu.cn

Received 29 November 2021; Accepted 22 February 2022; Published 5 April 2022

Academic Editor: Ahmed Mostafa Khalil

Copyright © 2022 Hang Xu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this work, we propose AGKN (attention-based graph learning kernel network), a novel framework to incorporate information of correlated firms of a target stock for its price prediction in an end-to-end way. We first construct a stock-axis attention module to extract dynamic and asymmetric spatial correlations through the kernel method and a graph learning module into which more accurate information can be integrated. An ensemble time-axis attention module is then applied to learn temporal correlations within each stock and market index. Finally, we utilize a transformer encoder to jointly attend to obtain information from different levels for correlations’ aggregation and prediction. Experiments with data collected from the Chinese stock market show that AGKN outperforms state-of-the-art baseline methods, making up to 4.3% lower error than the best competitors. The ablation study shows that AGKN pays more attention to hidden correlation between stocks, which improves model’s performance greatly.

1. Introduction

Stock prediction has drawn a lot of attention from researchers for a long time. Accurate stock prediction can help investors make appropriate decisions, acting as a meaningful index of portfolio allocation [1]. Efficient market hypothesis (EMH) assumes that all the participants in the market are informationally efficient and that all deals are traded at fair values [2], which mean that the stock market can respond to new information rapidly and stock prices are full representations of market. Therefore, it is impossible to predict stock prices under EMH’s framework. However, the rationality of this framework is questionable in reality. Behavioral economists then propose adaptive market hypothesis (AMH) to revise EMH [3]. AMH takes interaction between market and participants into consideration and indicates that it is feasible to obtain excess return through information asymmetry, which implies that the stock is predictable.

Traditional time-series forecasting methods include the AR model, the ARMA model, and the ARIMA model which take single time series as input and make prediction through fitting. Those models can only extract the linear relationship of time series, but the correlation in financial data is always nonlinear [4]. Recently, machine learning and deep learning have become gradually popular in stock prediction. Lee et al. use SVM to extract features and make prediction [5]. Ballings et al. forecast stock price based on ensembles’ algorithm including random forest, Adaboost, and kernel factory [6]. Such methods for individual prediction focus on temporal prices pattern of each stock, ignoring the interaction between stocks.

Some previous works utilize models based on LSTM and Markov random fields (MRF) for correlated stocks prediction [7, 8]. Qin et al. propose dual-stage attention-based recurrent neural network (DA-RNN) to learn relationships between stocks automatically [9]. Yoo et al. propose data-axis transformer with the multilevel context (DTML) model for stock movement prediction [8]. DTML utilizes attention-based LSTM and multihead attention for time-axis and data-axis relation extraction, respectively. However, the data-axis module follows time axis, which means that time-axis attention will influence model performance greatly. This will lead to unstable results and poor generalization ability of the model. One of the key challenges for multiple stocks prediction is “how to correlate historical price data both in stock axis and time
axis”. Many researchers choose to use graphing neural networks (GNNs) to address this issue because GNN can effectively extract correlation from nodes [7, 10, 11]. However, GNN has the following limitations. First, it requires graph structure as prior knowledge, which is always not defined in financial data. Second, those predefined structure between stocks is fixed, while true correlations keep changing with market. In order to address these issues, the kernel method-based model has been applied. The kernel method plays an important role for data presentation prior to neural network [12]. After mapping features such as vectors and matrix to high-dimensional space by the kernel method, the well-honed linear model can also achieve high performance. This method has a great advantage: it is nonparametric, which is crucial for computation complex. And it can enhance the expressiveness of raw data and extract useful information from mass data. And using correlated stocks for prediction may introduce too many parameters, as well as information noise. We take advantage of the kernel method to solve those problems.

In this study, we propose an end-to-end framework, AGKN (attention-based graph learning kernel network), for correlated stocks prediction. As demonstrated in Figure 1, our model consists of three parts: stock-axis attention generation module, time-axis attention generation module, and attention aggregation module. In stock axis, correlated stocks’ data are projected into a reproducing kernel Hilbert space (RKHS) with the learnable kernel method. A novel adjacent matrix extraction layer is further proposed for stock-axis attention capture based on transformed data. In time-axis, attention-based bidirectional LSTM (A-BiLSTM) is applied to obtain the attention scores. Ultimately, an aggregation module is used for attention scores’ aggregation and final prediction. In AGKN, asymmetric and dynamic attention scores are generated by the adjacent matrix extraction layer, avoiding the given disadvantages of GNN, and market data are included to respond to changes in macro-environment flexibly. The kernel method acts as a data intensifier and significantly improves model performance.

Our contributions are summarized as follows:

(i) To our best knowledge, this is the first attempt to study correlated stock prediction based on the kernel method. Because too many features as well as noises exist for correlated stocks’ prediction, this nonparametric method can enhance raw data and reduce model parameters to prevent overfitting.

(ii) We proposed a novel framework, AGKN, for correlated stocks’ prediction in an end-to-end way, which can obtain dynamic data relationships both in stock axis and time axis.

(iii) We conducted extensive experiments to evaluate AGKN. Experimental results show that AGKN outperforms the state-of-the-art methods. Furthermore, we analyze the stock-axis attention scores, and it proves that our novel architecture is effective in correlation extraction.

The reminders of this paper are organized as follows. In Section 2, we summarize relevant works in correlated stock prediction and kernel method. In Section 3, we present the main methodologies and components of AGKN. In Section 4, we show the experiment settings, results, and analysis. The conclusion will be given in Section 5.

2. Related Works

2.1. Graph Neural Network and Attention Mechanism for Correlated Stock Prediction. Most stocks can be clustered according to industries [13]. Stocks in the same industry have similar price trends in the long term except short-term random disturbance, which acts as a reliable hint for investors. However it remains a challenge to obtain correlation from historical data. Chen et al. take common shareholders in different stocks as nodes and use graph convolutional network (GCN) to capture correlation, which cannot fully represent real-time market changes due to lags of shareholder information update [11]. Cao et al. combine graph Fourier transformer and discrete Fourier transformer in spectral temporal graph neural network (stem-GNN) and use stem-GNN to model interseries and intraseries correlation [10]. Li et al. encode lead-lag relationships between stocks using weighted higher-order Markov random fields (MRFs) and treat them as nodes of MRFs [7]. However, they all require graph structure as prior knowledge, which is not defined in financial data. In fact, spatial-temporal graph neural networks are probably the most suitable type of GNN for multivariate time series, which takes multivariate time series and an external graph as input [14]. In addition, using graph attention network (GAT) for node-level and semantic-level relation capture has achieved good results [15, 16]. Wu et al. propose graph learning module to automatically extract the undirected relations among multivariate data. A novel mix-hop propagation layer and dilated inception layer are further applied to capture the spatial and temporal dependencies within each time series [17]. However, because financial data are always high noise, frequent alternations of those modules in the proposed framework may exacerbate noise propagation to reduce the model accuracy.

In order to make full use of correlated stocks data, we propose a novel framework to obtain both temporal and spatial attention from raw data in a synchronized rather than step-by-step way. Because financial data are marked by leptokurtic [18], volatility clustering [19], and low signal-noise ratio (SNR) [20], we need to design dedicated modules for specific tasks based on raw data.

2.2. Kernel Method in Neural Networks. The kernel method is closely related to infinitely wide neural networks. Neal and R.M. point out that a single hidden layer neural network with independent, identically distributed random parameters, in the limit of infinite width, is a function drawn from a Gaussian process (GP) [21]. Recent works extend it to deep neural networks and convolutional neural network (CNN) [22, 23]. Jacob et al. believe that least squares regression based on neural tangent kernel (NTK) equals to infinitely wide neural networks [24]. Lee et al. put forward that optimizing the last layer of an infinite wide neural network is equivalent
to a Gaussian process based on neural network [25]. Therefore, the kernel method can be applied as one way to improve the performance of deep neural networks. Mairal et al. propose convolutional kernel network (CKN) to learn to approximate the kernel feature map on training data and present the backpropagation method in CKN, bridging a gap between the kernel method and the neural network [26, 27]. Chen et al. generalize convolutional kernel networks to graph-structured data by representing graphs as a sequence of kernel feature maps. However, none of them focus on financial data, which have distinctive characters. Liu et al. propose the recurrent convolutional neural kernel (RCNK) model, which learns complementary features from historical data and text data, to predict the stock price movement [28]. RCNK constructs an explicit kernel mapping layer to replace fully connected layers to reduce the number of parameters and maintain accuracy. However, RCNK fails to take correlation between stocks into consideration, which is an important character in financial domain.

Generally, a positive defined kernel implicitly induces reproducing kernel Hilbert space (RKHS) and transforms the linear model in RCKS to nonlinear model in input space. This method can reduce the number of parameters in architecture and enhance data presentation. Therefore, merging kernel method and deep neural network can decrease the computation complexity and the risk of overfitting. Besides, through transformation of the learnable kernel method in network, information hidden in raw data is apt to be extracted in the following modules to improve model accuracy.

### 3. Proposed Approach

#### 3.1. Overview

AGKN is based on the kernel method and GNN, in which correlated stocks' information is used coherently to improve model performance. As illustrated in Figure 1, AGKN consists of stock-axis attention module, time-axis attention module, and aggregation module. To discover hidden correlation among stocks, our stock-axis attention module utilizes a novel layer to extract adjacency matrix. Prior to it, we use the learnable kernel method, instead of a fixed one, to transform data, acting as a data intensifier. As a result, the data intensifier can adapt itself to uncertain market circumstances. The learnable way makes it easy to extract correlation, which addresses the limitations of previous works concluded in Section 2.1. Figure 2 presents the framework of stock-axis attention module. In time axis, attention-based LSTM is used to capture relation within historical data of each stock. Instead of proceeding in lockstep, the operations in both directions are synchronized. It is because financial data are always low-SNR (signal-to-noise ratio) and noise tends to accumulate and propagate in feedforward. The synchronized structure can extract more useful context from raw data and reduce noise generation at the same time, which avoids drawbacks we put forward in Section 2.2. Finally, two attention maps are fed into output module, in which we apply transformer encoder [12, 20] for attention aggregation and a fully connected layer for prediction. In this way, temporal and spatial correlations are effectively combined and AGKN performance advances greatly. In more detail, the core components of our model are illustrated in the following.

#### 3.2. Convolutional Kernel Layer

We first elaborate on the general theory of kernel convolution (kervolution) developed in [26]. The kervolution learns to map the given data into a reproducing kernel Hilbert space (RKHS) with trainable parameters in neural network. In RKHS, we can find a subspace to divide transformed data. Given a set $\chi$, a positive definite kernel $K: \chi \times \chi \rightarrow \mathbb{R}$, and a mapping $\phi: \chi \rightarrow \mathcal{H}$:

![Figure 1: The overall architecture of AGKN, which consists of three main modules. First, stock-axis attention module computes the adjacent matrix $M_{out}$. Second, several A-BiLSTMs extract temporal attention within each stock and concatenate them as $T_{out}$. Finally, transformer encoder is used to aggregate them and make prediction.](image.png)
Specifically, consider two feature matrices \( \text{NLP} \) when polynomial kernel performs well for natural language processing.

For computational complexity while focusing on different levels as much as possible, 

\[
\begin{align*}
\text{adjacent matrix} & \rightarrow \text{ensemble stock-axis attention module} \\
\text{product kernel} & \rightarrow \text{Gaussian and polynomial kernel convolution}
\end{align*}
\]

where \( \| \cdot \|_2 \) denote unit \( \ell_2 \)-norm and \( \alpha \) is a hyperparameter. From Taylor expansion, RBF extends the kernel method to infinite dimensions:

\[
K_G(x, w) = \exp \left( \frac{\alpha}{2} \left( \|x\|_2^2 + \|w\|_2^2 \right) \right) \sum_{i=0}^{\infty} \left( x^T w \right)^i.
\]  

### 3.3. Stock-Axis Attention Generation

Some graph-based prediction models follow three main steps [10]. First, historical stock prices are fed into the sequence layer to output node embedding for each stock. These outputs are used to capture short-term price trend. Then, an adjacency matrix is created based on some other open graph sources such as Wiki-data or semantic text data on stock forum. Finally, a graph neural network is employed to combine the node embeddings and adjacency matrix as well as to make prediction. Such predefined and multidata-based models are likely to confuse information hidden in stock market and fail to capture dynamic correlation between stocks in different time steps [32]. These drawbacks restrict effective spatial dependencies learning of models. In this phase, we propose a novel matrix extraction layer to extract dynamic spatial dependencies among nodes (stocks) based on transformed data with the kernel method. Furthermore, influence between stocks is usually asymmetric. For instance, small firms’ stock prices tend to be more responsive to changes in other stocks prices than large firm does [33]. Thus, the influence of large firm’s stock price on small firm is relatively greater than reverse influence. Our data-driven method addresses these problems, extracting dynamic and asymmetric adjacent matrix automatically, which is illustrated as follows:

\[
M_1 = \tanh(EX),
\]

\[
M_2 = \text{SoftMax} \left( \tanh \left( M_1B_{i}^T \right) \right),
\]

\[
M_{out} = \alpha M_2 + (1 - \alpha)P,
\]
where $X$ denotes feature matrix transformed by kernel method, $E$, $B$, and $\alpha$ are learnable parameters, and $P$ represents Pearson correlation coefficient matrix.

Equation (7) is expanded as

$$M_1[i, j] = \tanh\left(\sum_{j=0}^{m} e_{i,j}x_{j,i}\right).$$  

(10)

From this perspective, items in matrix $M_1$ can be viewed as weight-average impact of other stocks and market on specific stock in specific time step. In this way, interactions among nodes are taken into consideration in each time step, which enhances the expressiveness of the model. The asymmetric property of our proposed method is achieved by equation (8), which summarizes information of all interacting nodes through hidden edges in entire rolling windows. $B$ acts as a weight matrix in this equation and SoftMax activation function is employed as regularize of the adjacent matrix.

### 3.4. Time-Axis Attention Generation

Another important part for price prediction is how to obtain temporal dependencies within historical data. To answer this question, we apply LSTM-based model. LSTM addresses the vanishing gradient problem and has stronger memory within time series. Given feature vectors and state vectors, a LSTM cell computes the current state vectors and feeds them into the next cell. Zhou et al. combined attention mechanism and LSTM to capture different information in a sequence, which can handle time series well so that we introduce it into AGKN [34]. The structure of this model is simple and easy to understand, which is shown below.

**Attention LSTM**: let $H$ be a matrix consisting of the hidden states in each time-step $[h_1, h_2, ..., h_T]$, where $T$ represents the sequence length, and the time-axis attention is computed as

$$M = \tanh(H),$$

$$\alpha = \text{SoftMax}(w^TM),$$

$$T_{out} = Ha^T,$$

where $w$ is a learnable parameter. The detailed implementation is shown in [34].

### 3.5. Multilevel Attention Aggregation

Finally, AGKN integrates stock-axis attention and time-axis attention for prediction. In this module, we employ multihead attention to tackle this task for its ability to jointly attend to information from different levels. Given attention matrices in two directions, $T_{out}$ and $M_{out}$, the input of attention $F$ is defined as

$$F = M_{out}T_{out}.$$

(12)

Thus, multihead attention learns to extract correlation between stocks from hybrid attention matrix $F$ and make final prediction.

Transformer encoder used for final prediction: the process of self-attention is shown as follows:

$$Q = FW_Q,$$

$$K = FW_K,$$

$$V = FW_V,$$

$$Z_0 = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where $d_k$ denotes sequence length of time series and $W_Q, W_K, W_V$ are learnable parameters.

In order to capture various information, multihead attention defines $N$ sets of $W_Q, W_K, W_V$ to obtain $N$ attention scores $Z_1, Z_2, ..., Z_N$. These scores focus on different aspects of data. The attention output is the average of all attention heads. Then, a residual connection is applied to avoid vanishing gradient:

$$Z = Z_{average} + F,$$

(14)

where $Z$ gives us some hints of correlation between stocks in an asymmetric way. Finally, several fully connected layers and activation functions are used for final prediction.

### 4. Experiments, Results, and Discussion

In this section, we present our experiment setting and results and give detailed analysis. We compare the performance of AGKN with other baseline models. Furthermore, to verify the importance of the kernel method and the adjacent matrix extraction layer in AGKN, we remove the corresponding modules and find that corresponding models perform poorly from different perspectives. Finally, we analyze the results from stock-axis attention generation to emphasize its function again.

#### 4.1. Dataset

We collect daily data of 13 stocks and 1 market index from Shanghai and Shenzhen stock markets in China from 3 January 2017 to 22 October 2021. The transaction data of each trading day are taken as raw data, including opening price, closing price, the highest price, the lowest price, trading volume, and amount. Table 1 summarizes statistics of dataset. The data are from https://tushare.pro/ (access on 1 November 2021).

We move a lag window with size of $h$ time steps to construct the feature matrix and label it with the next day’s closing price. We segment data with the first 70% of the date for the training dataset and final 30% for test data.

#### 4.2. Comparative Models

Firstly, to evaluate the effectiveness of stock and time attention module, we take AGKN with other baselines (KNN) [31],
A-BiLSTM [34], and long- and short-term time-series network (LSTNet) [35] will be used as baseline models for comparative experiments. Table 2 shows the abbreviations of comparative models.

4.3. Metrics and Hyperparameters. We evaluate the result of stock price prediction by two metrics: mean squared error (MSE) and root mean square percentage error (RMSPE). These two metrics are always used to measure the quality of regression problems, which is defined as follows:

\[
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_{i}^{\text{pre}} - y_{i}^{\text{true}})^2,
\]

\[
\text{RMSPE} = \left( \frac{1}{N} \sum_{i=1}^{N} \frac{(y_{i}^{\text{pre}} - y_{i}^{\text{true}})^2}{y_{i}^{\text{true}}} \right)^{1/2},
\]

where \(y_{i}^{\text{pre}}\) is predicted price and \(y_{i}^{\text{true}}\) is true price.

After attempting many times, AGKN’s hyperparameters are shown as follows. Lag window size \(h\) is 20, Gaussian kervolution’s kernel size is 4, polynomial kervolution’s \(d_{p}\) is 4, the number of epochs is 500, the number of A-BiLSTM’s layers is 2, and optimizer is Adam optimizer with both learning rate and regularization weight being 0.01.

4.4. Results. We test AGKN and its competitors. The performance is evaluated by MSE and RMSPE. Table 3 shows the performance of the different models, and Figure 4 visualizes the predicted price.

4.5. Ablation Study. From the results, we can conclude that

(1) Each module improves the model performance, and AGKN with complete structure produces the best results. AGKN outperforms AGKN-G and AGKN-S greatly, which means that G and S are more important in the whole framework. As for AGKN-K, AGKN do not show a huge improvement from it. Actually, as we can see from Figure 4, AGKN and its derivatives all perform bad in the area within red box of AGKN’s and AGKN-K’s result figures. However, in this area, there is an obvious synergetic effect between predicted and realistic trend for the AGKN model. It means that the kernel method we utilized can make the model pay more attention to hidden information which cannot be extracted with other model. k!_hatis what we expected the kernel method to do! AGKN-T’s performance does not differ greatly from AGKN. However, its result shows distinctive timelag. In other words, the predicted price is close to its prices of previous few days, which is not what we want. Actually, the time lag is as it should be because T module is used to extract time correlation. Without this module, time-axis information cannot be used effectively, causing the time lag.

(2) AGKN outperforms A-BiLSTM and LSTNet, which means it is far from enough to focus only on temporal correlation within historical data. We need to pay more attention to spatial dependencies. On the contrary, KNN’s performance is poor. It is because KNN is of numerical instability. If we apply it alone, we will always fail on training.

(3) Figure 5 shows the concatenated attention scores between stocks in Table 1 with respect to target stock (600276.SH). The scores represent how much target stock considers others at each moment. From the figure, we can observe three main points. First, certain stock scores change smoothly over time. This is natural because stock market is continuous so that the property of a stock does not change instantly at a moment. Second, attention scores vary greatly among stocks both at single time step and in the whole time period. 300122.SZ, 300244.SZ, 000931.SZ, and 603883.SH are especially crucial for target stock. The variety stresses the significance of correlated stocks prediction. Third, attention scores
Table 2: Model description.

| Abbreviation | Description |
|--------------|-------------|
| AGKN         | Attention-based graph kernel network |
| AGKN-K       | AGKN without the kernel method (K) |
| AGKN-G       | AGKN without the graphing learning method (G) |
| AGKN-S       | AGKN without stock-axis attention (S) |
| AGKN-T       | AGKN without time-axis attention (T) |
| A-BiLSTM     | An attention-LSTM model (state-of-the-art model) [34] |
| LSTNet       | A deep neural network, which combines convolutional and recurrent neural network [35] |
| KNN          | A model which integrates the kernel method and deep neural networks [31] |

Table 3: Experiment results.

| Model     | MSE      | RMSPE  |
|-----------|----------|--------|
| AGKN      | 17.4042  | 0.0480 |
| AGKN-K    | 19.1892  | 0.0492 |
| AGKN-G    | 46.8264  | 0.0786 |
| AGKN-S    | 19.7096  | 0.0580 |
| AGKN-T    | 17.7034  | 0.0482 |
| A-BiLSTM  | 44.4000  | 0.0915 |
| LSTNet    | 34.3967  | 0.0915 |
| KNN       | 1889.5324| 0.5664 |

Figure 4: Continued.
Figure 4: Continued.
for one stock differ greatly in different periods with large gap. This is because stocks are always volatility clustering and their correlations always change greatly along with time, which corresponds to dynamic character we emphasized above.

5. Conclusions

In this study, we proposed a hybrid model named attention-based graph learning kernel network (AGKN) for correlated stock prediction. AGKN effectively extracts attention both in time axis and data axis without any prior knowledge. By mapping raw data into RCHKs, we proposed a novel way to exploit the correlations among multiple stock data. The experiment results prove the effectiveness of each section in AGKN. Furthermore, our model outperforms a variety of stock prediction methods and acts as a development of the multivariate stock movement prediction in [8].

Data Availability

The data used to support the findings of this study are available at https://tushare.pro/.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.
References

[1] W. Long, Z. Lu, and L. Cui, “Deep learning-based feature engineering for stock price movement prediction,” Knowledge-Based Systems, vol. 164, no. 15, pp. 163–173, 2019.

[2] E. F. Fama, “The behavior of stock-market prices,” Journal of Business, vol. 38, no. 1, pp. 34–105, 1965.

[3] A. W. Lo, “Reconciling efficient markets with behavioral finance: the adaptive markets hypothesis,” The Journal of Investment Consulting, vol. 7, no. 2, pp. 21–44, 2005.

[4] H. Lütkepohl, New Introduction to Multiple Time Series Analysis, Springer Science & Business Media, Berlin/Heidelberg, Germany, 2005.

[5] M.-C. Lee, “Using support vector machine with a hybrid feature selection method to the stock trend prediction,” Expert Systems with Applications, vol. 36, no. 8, pp. 10896–10904, 2009.

[6] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryn, “Evaluating multiple classifiers for stock price direction prediction,” Expert Systems with Applications, vol. 42, no. 20, pp. 7046–7056, 2015.

[7] C. Li, D. Song, and D. Tao, “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, pp. 1141–1151, New York, NY, USA, 2019.

[8] J. Yoo, Y. Soun, Y. C. Park, and U. Kang, “Accurate multivariate stock movement prediction via data-Axis transformer with multi-level contexts,” in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2037–2045, Singapore, 2021.

[9] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” 2017, https://arxiv.org/abs/1704.02971.

[10] D. Cao, Y. Wang, J. Duan et al., “Spectral temporal graph neural network for multivariate time-series forecasting,” 2021, https://arxiv.org.abs/2103.07719.

[11] Y. Chen, Z. Wei, and X. Huang, “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, pp. 1655–1658, Toronto Italy, 2018.

[12] B. Scholkopf and A. J. Smola, Learning with kernels: support vector machines and other kernel-based learning methods, MIT Press, Cambridge, MA, USA, 2001.

[13] B. S. Bini and T. Mathew, “Clustering and regression techniques for stock prediction,” Procedia Technology, vol. 24, pp. 1248–1255, 2016.

[14] M. Li and Z. Zhu, “Spatial-temporal fusion graph neural networks for traffic flow forecasting,” 2020, https://arxiv.org/abs/2012.09641.

[15] X. Wang, H. Ji, C. Shi et al., “Heterogeneous graph attention network,” in Proceedings of the World Wide Web Conference, pp. 2022–2032, San Francisco, CA, USA, 2019.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” 2017, https://arxiv.org/abs/1710.10903.

[17] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, “Connecting the dots: multivariate time series forecasting with graph neural networks,” in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 753–763, New York, NY, USA, 2020, August.

[18] B. B. Mandelbrot, “The variation of certain speculative prices,” in Fractals and Scaling in Finance, pp. 371–418, Springer, New York, NY, USA, 1997.

[19] T. Lux and M. Marchesi, Volatility clustering in financial markets: a microsimulation of interacting agents, International Journal of Theoretical and Applied Finance, vol. 3, no. 4, pp. 675–702, 2000.

[20] J. B. De Long, A. Shleifer, L. H. Summers, and R. J. Waldmann, “Noise trader risk in financial markets,” Journal of Political Economy, vol. 98, no. 4, pp. 703–738, 1990.

[21] R. M. Neal, “Priors for infinite networks,” in Bayesian Learning for Neural Networks, pp. 29–53, Springer, New York, NY, USA, 1996.

[22] A. G. D. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, “Gaussian process behaviour in wide deep neural networks,” 2018, https://arxiv.org/abs/1804.11271.

[23] R. Novak, L. Xiao, J. Lee et al., “Bayesian deep convolutional networks with many channels are Gaussian processes,” 2018, https://arxiv.org/abs/1810.05148.

[24] A. Jacot, F. Gabriel, and C. Hongler, “Neural tangent kernel: convergence and generalization in neural networks,” 2018, https://arxiv.org/abs/1806.07572.

[25] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, “Deep neural networks as Gaussian processes,” 2017, https://arxiv.org/abs/1711.00165.

[26] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, “Convolutional kernel networks,” Advances in Neural Information Processing Systems, vol. 27, pp. 2627–2635, 2014.

[27] J. Mairal, “End-to-end kernel learning with supervised convolutional kernel networks,” Advances in Neural Information Processing Systems, vol. 29, pp. 1399–1407, 2016.

[28] S. Liu, X. Zhang, Y. Wang, and G. Feng, “Recurrent convolutional neural kernel model for stock price movement prediction,” PLoS One, vol. 15, no. 6, Article ID e0234206, 2020.

[29] L. Bo, K. Lai, X. Ren, and D. Fox, “Object recognition with hierarchical kernel descriptors,” in Proceedings of the Computer Vision & Pattern Recognition, June 2011.

[30] J. Dai, H. Qi, Y. Xiong et al., “Deformable convolutional networks,” IEEE, 2017, https://arxiv.org/abs/1703.06211.

[31] C. Wang, J. Yang, L. Xie, and J. Yuan, “Kervolutional neural networks,” 2019, https://arxiv.org/abs/1904.03955.

[32] A. Silvernoinen and S. Thorp, “Financialization, crisis and commodity correlation dynamics,” Journal of International Financial Markets, Institutions and Money, vol. 24, pp. 42–65, 2013.

[33] Y.-W. Cheung and L. K. Ng, “Stock price dynamics and firm size: an empirical investigation,” The Journal of Finance, vol. 47, no. 5, pp. 1985–1997, 1992.

[34] P. Zhou, W. Shi, J. Tian, Z. Qi, and B. Xu, “Attention-based bidirectional long short-term memory networks for relation classification,” in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 2, Berlin, Germany, August 2016.

[35] G. Lai, W. C. Chang, Y. Yang, and H. Liu, “Modeling long-and short-term temporal patterns with deep neural networks,” in Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 95–104, Ann Arbor, MI, USA, 2018, June.