Equity2Vec: End-to-end Deep Learning Framework for Cross-sectional Asset Pricing

Qiong Wu
William & Mary
qwu05@email.wm.edu

Christopher G. Brinton
Purdue University
cgb@purdue.edu

Zheng Zhang
William & Mary
zzhang14@email.wm.edu

Mihai Cucuringu
University of Oxford
The Alan Turing Institute
mihai.cucuringu@stats.ox.ac.uk

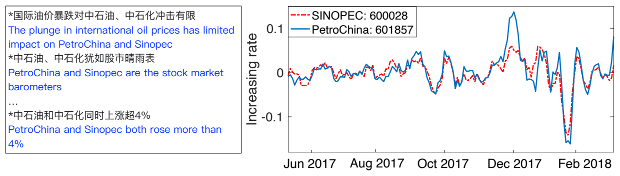
Andrea Pizzoferrato
University of Bath
The Alan Turing Institute
ap2873@bath.ac.uk

Zhenming Liu
William & Mary
zliu20@wm.edu

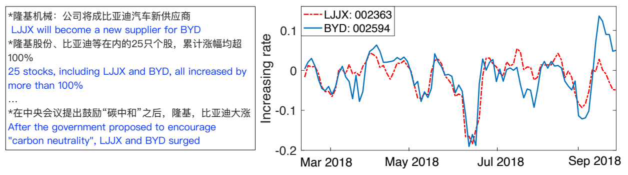
ABSTRACT

Pricing assets has attracted significant attention from the financial technology community. We observe that the existing solutions overlook the cross-sectional effects and not fully leveraged the heterogeneous data sets, leading to sub-optimal performance.

To this end, we propose an end-to-end deep learning framework to price the assets. Our framework possesses two main properties: 1) We propose Equity2Vec, a graph-based component that effectively captures both long-term and evolving cross-sectional interactions. 2) The framework simultaneously leverages all the available heterogeneous alpha sources including technical indicators, financial news signals, and cross-sectional signals. Experimental results on datasets from the real-world stock market show that our approach outperforms the existing state-of-the-art approaches. Furthermore, market trading simulations demonstrate that our framework monetizes the signals effectively.


*国际油价暴跌对中石油、中石化冲击有限
The plunge in international oil prices has limited impact on PetroChina and Sinopec
*中石油、中石化犹如股市晴雨表
PetroChina and Sinopec are the stock market barometers
...
*中石油和中石化同时上涨超4%
PetroChina and Sinopec both rose more than 4%

(a) The co-mentions capture sector relation between stocks


*隆基机械：公司将成比亚迪汽车新供应商
LJJX will become a new supplier for BYD
*隆基股份、比亚迪等在内的25只个股，累计涨幅均超100%
25 stocks, including LJJX and BYD, all increased by more than 100%
...
*在中央会议提出鼓励“碳中和”之后，隆基，比亚迪大涨
After the government proposed to encourage "carbon neutrality", LJJX and BYD surged

(b) The co-mentions capture supply-chain relation between stocks

Figure 1: Examples showing our key observations: When the news mention stocks frequently, the stocks are 1) likely to reflect relations, such as sector and supply-chain, 2) likely to have similar movement on prices.

1 INTRODUCTION

It is widely acknowledged that forecasting stock prices is a difficult task. Most traditional efforts rely on time series analysis models, such as Autoregressive models [34], Kalman Filters [43], and technical analysis [31, 44, 45]. Deep neural networks, especially recurrent neural networks (RNN) [10, 11, 20, 26, 51, 57, 61] recently emerged as an effective solution for stock prediction tasks. Such lines of work have significantly increased in popularity in recent years, mostly fueled by the fact that a large collection of high-quality financial data sets have become available.

We observe two fundamental limitations in the prior works: *1. Cross-sectional effects are not properly leveraged.* Most existing approaches treat each stock independently and overlook the cross-sectional effect. The cross-sectional effect posits the fact that the information from one stock may influence/impact another stock's price change in both static and dynamic aspects. Statically, the stocks that share the same intrinsic properties may move synchronously. For instance, when Twitter goes up, Facebook is more likely to go up because they are in the same sector (social media advertising). Dynamically, stock relation is also temporally evolving. For example, in early 2021, AMC theatres, GameStop, and BlackBerry suddenly exhibit co-movement driven by investors on social media who are buying up these stocks.

2. Heterogeneous data sets are not leveraged to their fullest extent. Most models use only one type of data (i.e., either textual information or "technical factors" [26, 41, 42, 48, 60] in numeric form). The latter is derived from prices and traded volumes. It remains open to building a model that leverages heterogeneous data sources. This model needs to reconcile and aggregate information from different data sources.

An efficient stock embedding scheme that addresses the first limitation must determine: 1) what data source includes the co-movement information, 2) how to extract the cross-sectional signals from the data source, and 3) how to incorporate both the static and dynamic stock relations into the stock embedding. We propose Equity2Vec that answers the three questions.

Ideally, the stock representation should reflect comprehensive relations such as sector, supply chain, value, growth, business cycles, volatility, and analysts' confidence towards the stock. One possible way is to collect such information manually from experts and analysts, but it is inefficient and costly. Further, there are no widely agreed upon standard approaches to converts people's opinions into stock representations. *Since millions of investors, analysts, and financial experts share opinions, events, comments, and transactions about stocks in the news, we consider using news as a data source to learn stock representations.*

Next, we make two key observations by analyzing news on stocks. When two stocks are frequently co-mentioned, 1) they are likely to share common characteristics such as sector and supply-chain relation, 2) their prices tend to have a similar trend. For example, the co-mentioned stocks in Figure 1 (a) are in the same sector (energy), while Figure 1 (b) shows they have a supply-chain relationship (i.e., LJJX is a supplier of BYD). In both cases, the prices of these co-mentioned stocks often move synchronously and most often in the same direction. *Based on our observations, we use news co-mention*[1] *to learn the stock representation.*

Moreover, EQUITY2VEC extracts the long-term (static) and evolving (dynamic) stock relations by the following approach. To capture the long-term relation, we build a global stock co-occurrence matrix [49] (see Figure 3(a) (❶)) with a long observation window. We extract the stocks' long-term representations via matrix factorization of the co-occurrence matrix. To learn the evolving relation, we build a stock graph that reflects the dynamic neighboring relations, where EQUITY2VEC propagates the embedding of a stock to its neighbors to capture the cross-sectional information of the stocks effectively.

To leverage the heterogeneous data sources (i.e., second limitation of prior works), we propose a framework that integrates the learned stock embeddings, news signals, and technical factors into a neural network model to make the final prediction. We perform extensive experiments on real-world data that contains more than 3,600 stocks in the Chinese stock market from 2009 to 2018. The experimental results demonstrate the effectiveness of stock representations extracted by EQUITY2VEC outperforms existing state-of-the-art works. The market trading simulation illustrates that EQUITY2VEC along with the proposed framework increases profit significantly.

In summary, this paper makes the following contributions:

- We propose EQUITY2VEC that incorporates news into stock embeddings. To the best of our knowledge, EQUITY2VEC is the first work that mines the stock representation from the news co-mention. Moreover, EQUITY2VEC captures both long-term (static) and evolving (dynamic) relations between stocks.
- We forecast stock prices using multiple categories of signals (i.e., heterogeneous data sets), including cross-sectional embeddings, technical signals, and financial news signals.
- Extensive experiments on real-world data sets confirm the efficacy of our approach, comparing favorably to state-of-the-art methods.

2 PRELIMINARIES AND FRAMEWORK OVERVIEW

Problem setting. Given a universe of n stocks $s_1, s_2, ..., s_n$, the stock price trend is log return for a given stock i on day t $r_t^i = \log\left(\frac{p_t^i}{p_{t-1}^i},\right)$ where p_t^i denotes the open price of stock i on day t. We formulate the task of predicting the future price trend as a regression problem. The response is the future return r_{t+1}^i, and $\mathbf{x}_t^i$ denotes the vector of features associated with stock i on day t. The historical features up to time t are defined as $\mathbf{x}_{\le t}^i$. We aim to learn the function $r_{t+1}^i = f(\mathbf{x}_{\le t}^i)$.

Framework Overview. Figure 2 shows our overall framework, which includes the EQUITY2VEC component and the heterogeneous data source component. The EQUITY2VEC component first learns the stocks' long-term relations from the global stock co-occurrence matrix, and then extracts the static stock embedding as e_i (❶). Then, at time t, we build the temporal graph G_t dynamically based on the local stock co-occurrence matrix to capture the evolving relations (❷). Within graph G_t, each solid circle represents a stock. Stocks close to one another are likely to be associated with a similar moving trend. Finally, our approach obtains the final stock representation c_t^i by propagating its neighbors' basic embedding via an attention mechanism (❸).

In the heterogeneous data source component, we integrate the stock embedding (c_t^i) with dynamic input (g_t^i) (generated from technical factors and online textual data), and denote the heterogeneous output as h_t^i. Finally, the RNN model forecasts future return r_{t+1}^i based on the input h_t^i.

3 EQUITY2VEC FROM NEWS

We propose EQUITY2VEC, which mines the stock embeddings from the news, since such data comprises a valuable knowledge repository with rich relation information between stocks from the crowd of financial experts/journalists. Our approach is inspired by the observation (as depicted in Figure 1) that stocks frequently co-mentioned by the same news are likely to share similar properties and exhibit co-movements in their price trends. We formulate the stock co-occurrence matrix, and use matrix factorization to extract the *static* stock representation in Section 3.1. To explicitly leverage the cross-sectional signals and circumvent the challenge that the relations between stocks are evolving, we build a temporal stock graph and fine-tune the stock representations by *dynamically* infusing neighbors latent representations (Section 3.2).

3.1 Capturing long-term stock relations

We now focus on learning the long-term relation, and discuss how to address the dynamic relations in the next subsection.

Co-occurrence Matrix. We build a stock co-occurrence matrix by counting the stock co-occurrence within each news. We prefer the stock co-occurrence matrix instead of the entire news and stock matrix for the following two reasons: (1) The size of the stock co-occurrence matrix only depends on the number of stocks, and is much smaller than the news and stock matrix. (2) We only care about the stock representation, which thus renders news representations as not necessary. We use the global co-occurrence matrix to obtain static representations that reflect the essential relations between stocks in the long term.

Formally, suppose we have n stocks and $\mathbf{X} \in \mathbf{R}^{n\times n}$ denotes the stock co-occurrence matrix. Figure 3(a) (❶) shows $\mathbf{X}_{i,j}$, counting the number of articles that mention both s_i and s_j before testing phase. We associate a vector $e_i \in \mathbf{R}^d$ to represent the stock i's static representation, where d is the dimension of the stock representation. In this way, the similarity between stock i and stock j can be formulated as the inner product of e_i and e_j, given by $e_i^\top e_j$.

Matrix Factorization. Figure 3(a) (❷) shows that we adopt matrix factorization [32] to learn the embedding of the stocks. Here, matrix

[1]Through the statistic analysis of real-world online media, a piece of news refers to 3.0 stocks on average.

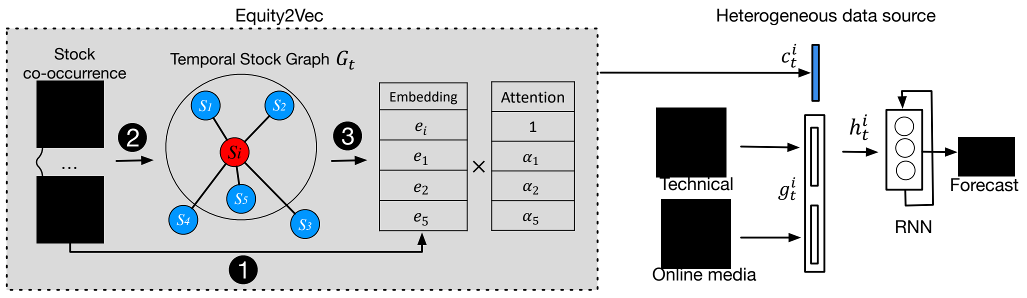


Figure 2: The illustration of our end-to-end framework. It contains the EQUITY2VEC component (Section 3) and heterogeneous data source component (Section 4).

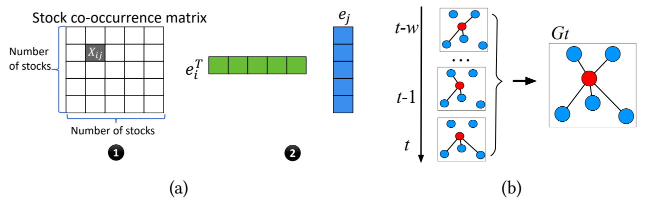


Figure 3: (a) Our approach to build the stock co-occurrence matrix and calculate the static embedding for stocks. (b) The construction of temporal graph.

factorization works as a collaborative filtering method. The idea behind matrix factorization is to learn the latent representation where stocks near each other will likely obtain similar embeddings. Given the co-occurrence matrix reflects the similarity between stocks, we obtain the latent representation of stocks through fitting the training data by optimizing the objective function

$$J_s = \sum_{i,j=1}^{n} (e_i^\top e_j - \mathbf{X}_{ij})^2. \tag{1}$$

In reality, there could exist prior bias of stocks as the prior preference of financial journalists/experts. Hence, we use b_i and b_j as the bias of stock i and stock j and introduce them to the objective function,

$$J_s = \sum_{i,j=1}^{n} (e_i^\top e_j + b_i + b_j - \mathbf{X}_{ij})^2 + \beta(||\theta||^2) \tag{2}$$

where $||\theta|| = (||e_i||^2 + ||e_j||^2) + b_i^2 + b_j^2$ and $\beta(||\theta||^2)$ is the regularization term that prevents overfitting.

3.2 Capturing evolving stock relations

The latent representation learned from the above global occurrence matrix reflects the static stock relations. Motivated by the fact that the stock relations are changing over time and cross-asset signals are beneficial towards stock price prediction, we further fine-tune the stock representation by building a temporal stock graph and infusing the neighbors' embedding dynamically.

Temporal stock graph. As shown in Figure 3(b), the construction of G_t consists of two steps. (1) Construct the stock graph using the co-occurrence matrix. Assume $\tilde{G}_t = \{\mathcal{V}, \mathcal{E}_t\}$ is the stock graph at time t, where $\mathcal{V} = \{s_1, \ldots, s_n\}$ is the set of stocks. $(s_i, s_j) \in \mathcal{E}_t$ if and only if s_i and s_j are co-mentioned by the news collected at time t. The edge eight over s_i and s_j is the number of co-occurrences across all news on date t. (2) Due to insufficient number of news about specific stocks in time t, we maintain a sliding window w (w is a hyper-parameter) to collect a sequence of stock graphs and then construct G_t by taking an exponential moving average of $\tilde{G}_{t-w}, \ldots, \tilde{G}_t, \tilde{G}_t$, where we assign a nearby graph a larger weight.

Propagation of neighbors' embedding via a stock attention mechanism. Given the temporal graph (G_t) identifies the dynamic stock structure, it is crucial to appropriately update the stock representation by infusing the current neighbors' embedding. Motivated by the fact that not all the neighbors contribute to the current stock trend, we filter out the stocks that are too far away from the current stock (See Section 6 for more details). Specifically, for stock s_i, we focus on the k nearest neighbors when sorting by the edge weight (which reflects the magnitude of the co-occurrence), where k is a hyper-parameter. Formally, let $S_t(i)$ denote the set of k nearest neighbors of s_i in G_t at time t.

We introduce an attention mechanism [58] to infuse the neighbors embedding weighted by an assigned attention value. In this way, we reward the stocks offering more forecasting power by assigning them larger attention values.

$$c_t^i = \sum_{j \in S_t(i)} \alpha_{ij} e_j, \tag{3}$$

$$\sum_{j \in S_t(i)} \alpha_{ij} = 1 \text{ for } j \in S_t(i), \tag{4}$$

where c_t^i denotes the fine-tuned stock representation for stock i at time t, and $\alpha_{i,j} \in \mathbf{R}$ is the attention weight on the embedding e_j, which is given by

$$\alpha_{ij} = \frac{exp(f(e_i, e_j))}{\sum_{l \in S(i)} exp(f(e_i, e_l))}. \tag{5}$$

The weights define which neighboring stocks are more significant. $f(e_i, e_j)$ measures the compatibility between embeddings e_i and e_j, and is parameterized by a feed-forward network with a single hidden layer, which is jointly trained with other parts of the model. We let $f(\cdot, \cdot)$ have the following functional form

$$f(e_i, e_j) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[e_i; e_j] + b_a), \tag{6}$$

where $\mathbf{v}_a$ and $\mathbf{W}_a$ are weight matrices, and b_a is the bias vector [58], obtained during model training via backpropagation.

4 LEVERAGE HETEROGENEOUS DATA SOURCES

In this section, we show how to integrate heterogeneous data sources for making forecasts, and how we gather different sources of signals.

4.1 Sequential modeling

Finally, we integrate the stock embedding from Equity2Vec with stock dynamic features into the neural net model to predict the future return. The stock dynamic input is given by g_t^i, which stems from technical factors and news data. We overlay the stock vector c_t^i with g_t^i as an input. Specifically, let h_t^i be the hidden state at time t for stock i

$$h_t^i = [c_t^i, g_t^i], \tag{7}$$

where $[\cdot, \cdot]$ denotes a direct concatenation.

Keeping in mind that stock trends are highly influenced by a variety of time-series market signals, it is intuitive to take the historical features of a stock as the most influential input to predict its future trend. Therefore, we use Recurrent Neural Networks [40, 53] as the neural net model. LSTM is a variant of the recurrent net, which is capable of learning long-term dependencies. The final output is given by

$$v_{t-T}^i, ..., v_{t-1}^i, v_t^i = \text{LSTM}(h_1^i, h_2^i ... h_T^i; \theta_l), \tag{8}$$

where θ_l denotes the parameters from LSTM.

Temporal attention layer. Since a stock's historical data contributes to its price trend unequally, we adopt the attention mechanism at the temporal level. We consider

$$\begin{aligned} r_{t+1}^i &= \textstyle\sum_p \beta_p v_p^i, \\ \beta_p &= \frac{exp(f(v_p^i, v_q^i))}{\sum_q exp(f(v_p^i, v_q^i))}, \end{aligned} \tag{9}$$

where β_p is the attention weight for prior date p indicating the importance of the date. We then compute the weighted sum to incorporate the sequential data and temporal attention.

Assume we have m trading days and n stocks. We use the mean squared error as the loss function for gradient descent, given by

$$J = \frac{1}{mn} \sum_{i=1}^{n} \sum_{t=1}^{m} (r_{t+1}^i - \hat{r}_{t+1}^i)^2. \tag{10}$$

Alg. 1 describes our entire algorithmic training pipeline. Specifically, Lines 2 & 3 show the procedure to extract long-term relations, and Lines 7-11 show the steps to extract the evolving relation.

4.2 Gathering difference sources of alphas

Financial studies have attributed stock movements to three types of market information, i.e., cross-sectional signals, numerical technical indicators, and news features. To the best of our knowledge, the proposed framework is the first one that fuses technical factors, financial news and stock embedding together for stock predictions.

Stock graph: leveraging cross-sectional signals. Trading on cross-sectional signals (i.e., when Google goes up, Facebook is more likely to go up) is remarkably difficult because we need to examine all possible relations. We leverage news articles that mention multiple stocks to detect correlations between stock prices. Detecting co-movements by news appears to be much more effective than existing methods. Our Equity2Vec learns both long-term and evolving relations.

Algorithm 1 Our Algorithmic Training Pipeline

```
Input: Online news corpus, technical factors
Output: Prediction on future $\hat{r}_{t+1}^i$
1: Variables: Stock embedding matrix $\mathbf{E}$, stock attention parameters $\alpha_{i,j}$, LSTM parameters $\theta_l$, temporal attention parameters $\beta_i$.
2: Build the global stock co-occurrence matrix $\mathbf{X}$.
3: Use the matrix factorization on $\mathbf{X}$ with the loss function (Equation 2) to extract static stock embedding.
4: repeat
5:   $s_i \leftarrow$ stock $i$ from universe
6:   for time stamp $t$ do
7:     Build temporal graph $G_t$
8:     for stock $j$ in $S_i$ do
9:       Obtain the $\alpha_{i,j}$ with Equation 5
10:    end for
11:    Calculate the stock $i$'s final representation $c_t^i$
12:    Concatenate with the stock dynamic input to obtain final representation $h_t^i$ with Equation 7
13:    Forecast future return $\hat{r}_{t+1}^i$ using Equation 9.
14:  end for
15:  Calculate prediction loss $J$ by Equation10
16:  Update parameters based on the gradient of $J$
17: until convergence
```

Technical factors: hand-built features are more effective. We note that features extracted by deep learning [12, 36] are often less effective than features (technical factors) crafted by financial professionals [9]. Thus, we overlay an LSTM over technical factors, so that we can simultaneously leverage expertise from financial professionals, and also extract serial correlations from deep learning models.

Financail news Advancing development of Natural Language Processing techniques has inspired increasing efforts on stock trend prediction by automatically analyzing stock-related articles [1, 8, 26]. We pre-trained Word2vec [38, 39] on the news corpus from the training data set to produce word embeddings. We average all the word vectors in a piece of news to represent the news vector. For a given date t and stock i, we compute the daily news vector by extracting the news related to stock i and averaging the news vectors within date t.

5 EVALUATION METHODOLOGY

We now evaluate the methodology proposed. We focus on the Chinese market, whose value is the second largest in the world.

5.1 Data Collection

Chinese Equity Market. Our data set consists of daily prices and trading volumes of approximately 3,600 stocks between 2009 and 2018. We consider the universe with all the stocks except for the very illiquid ones. We use open prices (at 9:30 am) to compute the daily returns, and we focus on predicting the next 5-day returns. The last three years of the period are out-of-sample.

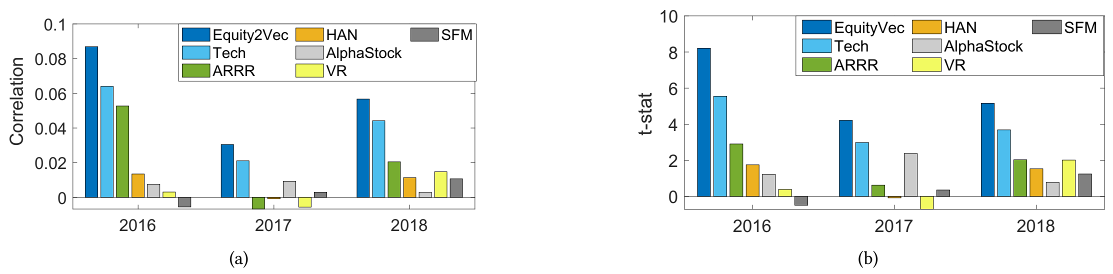


Figure 4: Performance comparison in terms of correlation (a) and t-statistic (b) among our Equity2Vec, ARRR, HAN, AlphaStock, VR, and SFM. For both correlation and t-statistic, higher scores are better.

Technical Factors. We manually build 337 technical factors based on the previous studies [2, 9, 23, 28, 50]. "Technical factor" is a broad term encompassing indicators constructed directly from data related to trading activities. Specifically, all these factors are derived from price and dollar volume by mathematical calculation. Table 1 shows a set of popular technical factors.

Table 1: A set of popular technical indicators and the corresponding description.

Factors	Description
EMA	Exponential moving average over price or dollar volume. [9]
RSI	The magnitude of one equity's recent price changes. [9]
ROC	Price variation from one period to the next. [19]
Volume Std	Standard deviation of volume. [18]
VCR	Volume cumulative return [9]

News Dataset. We crawled all the financial news between 2009/01/01 and 2018/08/30 from a major Chinese news website Sina[2]. It has a total number of 2.6 million articles. Each article can refer to one or multiple stocks. On average, a piece of news refers to 2.94 stocks. We link each of the collected news articles to a specific stock if the news mentions the stock in the title or content. The timestamps of news published online are usually unreliable (the dates are reliable, but the hour or minute information is usually inaccurate). We use news signals on the next trading day or later to avoid look-ahead issues.

5.2 Experimental settings

Training and Testing Data. We use three years of data for training, one year of data for validation, and one year for testing. The model is re-trained every testing year. For example, the training set starts from Jan 1, 2012 to Dec 31, 2014. The corresponding validation period is from Jan 15, 2015, to Dec 16, 2015. We use the validation set to tune the hyperparameters and build the model. Then we use the trained model to forecast returns of equity in the same universe from Jan 1, 2016 to Dec 31, 2016, where we set 10 trading days as the "gap". We set a "gap" between training and validation periods, and validation periods and testing periods to avoid look-ahead issues. The model is then re-trained by using data in the second training period (2013 to 2015) to make forecast on the second testing year.

Parameter Setting. We use the standard grid search to select the hyper-parameters in our experiments. We build the global co-occurrence matrix in Section 3.1 by using all the news before the first day of the testing year. To learn the stocks' embedding, we tune the dimension of stocks' representation within {32, 64, 128, 256}. We explore the number of LSTM cell within {2, 5, 10, 20}. We greedily search the number of neighbors for the stock graph from no neighbors to all the neighbors. The sliding window for temporal graph w is tuned within {2, 5, 10, 20, 60}. In addition, we tune the learning rate within {0.001, 0.01} with the Adam optimizer [30], and set the batch size within {128, 256}.

Evaluation Metrics. We evaluate our performance in terms of correlation, t-statistic, and PnL.

Correlation. Unlike the other regression tasks, correlation [4] is a preferable metric in stock price prediction compared to MSE since the direction instead of magnitude is more crucial for forecasting return.

Significance test (t-statistics). The use of t-statistics estimators [47] can account for the serial and cross-sectional correlations. Recall that $\mathbf{r}_t \in \mathbf{R}^n$ is a vector of responses and $\hat{\mathbf{r}}_t \in \mathbf{R}^n$ is the forecast of a model to be evaluated. We examine whether the signals are correlated with the responses, i.e., for each t we run the regression model $\mathbf{r}_t = \beta_t \hat{\mathbf{r}}_t + \epsilon$, and test whether we can reject the null hypothesis that the series $\beta_t = 0$ for all t. Note that the noises in the regression model are serially correlated, and thus we use the Newey-West [47] estimator to adjust for serial correlation issues.

PnL. Profit & Loss (PnL) is a standard performance measure used in trading. PnL captures the total profit or loss of a portfolio over a specified period. The PnL of all forecasts made on day t is given by

$$\text{PnL} = \frac{1}{n}\sum_{i}^{n} sign(\hat{r}_t^i) * r_t^i, \quad t = 1, \dots, m, \tag{11}$$

5.3 Baselines for comparison

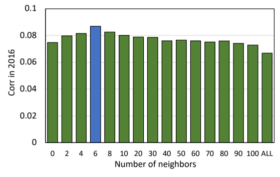
To test our proposed deep learning framework, we compare our model against state-of-art baselines, as described below. For all the baselines, we use the validation data set to configure the hyper-parameters.

SFM [61]. SFM is designed for stock prediction. It decomposes the hidden states of an LSTM [51] network into multiple frequencies by using Discrete Fourier Transform (DFT) in order for the model to capture signals at different horizons.

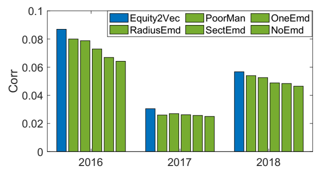
HAN [26]. This work introduces a so-called hybrid attention technique that translates news into signals. HAN uses Word2Vec to transfer news into vectors, and uses RNN as the modeling method.

AlphaStock [59]. AlphaStock integrates deep attention networks reinforcement learning with the optimization of the Sharpe Ratio. For each stock, AlphaStock uses LSTM [52] with attention on hidden states to extract the stock representation. Next, it relies on CAAN, a self-attention layer, to capture the inter-relations among

[2]https://finance.sina.com.cn



(a) *Varying k in testing year 2016.*



(b) *Different embedding methods.*

Figure 5: The effects of using different number of neighbors and effects of learned stock representation.

stocks. We implement LSTM with basic CAAN, and change the forecast into returns, instead of winning scores.

Vector Autoregression. We include a standard linear vector autoregression (VAR) [46]. VAR is a typical stock forecast baseline. Formally, it assumes $\mathbf{r}_{t+1} = f(\mathbf{x}_t) + \xi_t$, where $\mathbf{x}_t$ denotes the features of all stocks, and $\mathbf{r}_{t+1} \triangleq (r_{t+1,1}, \ldots, r_{t+1,n})$ denotes the future return of all stocks in the universe.

ARRR [60]. ARRR is a new regularization technique designed to address the overfitting issue in vector autoregression under the high-dimensional setting. Stock prediction is one of its applications. Specifically, ARRR involves two SVD steps; the first SVD is for estimating the precision matrix of the features, and the second SVD is for solving the matrix denoising problem.

6 PERFORMANCE AND DISCUSSION

In this section, we discuss our overall performance, analyze the effectiveness of k-nearest neighbors, the learned stock embedding, and market trading simulation results.

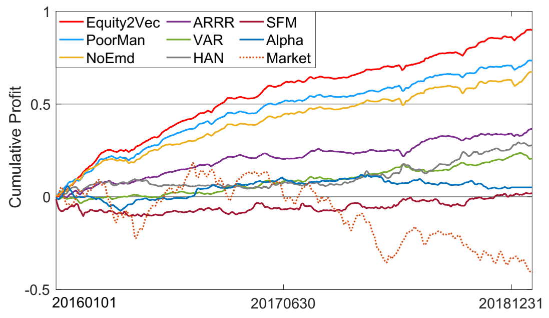
Overall Performance on Correlation and t-statistic. Figure 4(a) and Figure 4(b) report the comprehensive analysis on all compared methods, for each testing year in terms of both correlation and t-statistic. The results confirm that our EQUITY2VEC method consistently outperforms all other baselines, for each testing year and across all metrics.

Impact of Different Number of Neighbors. We investigate the number of neighbors k against the correlation metric. Figure 5(a) shows that with the increase of k, the out-of-sample correlation first increases and then decreases. This indicates the performance gains from our choices on k-nearest neighbors graph in Section 3.2.

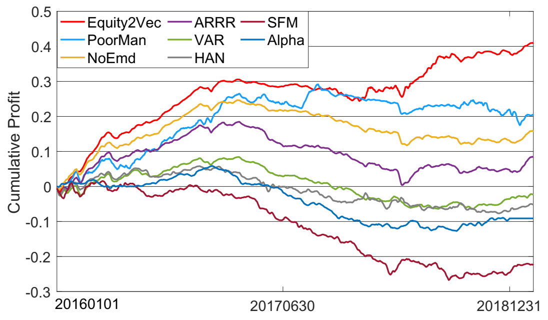
Effect of Learned Stock Representation. To demonstrate the effects of learned stock representations for stock prediction, we investigate the performance of EQUITY2VEC by replacing the learned stock representations with the following representations:

- NoEmd: Remove the EQUITY2VEC module.
- PoorMan: Remove the influence from neighbors.
- SectEmd: Replace the embedding with the aggregated embedding from the same sector.
- OneEmd: Replace the embedding with the embedding from all the neighbors.
- RadiusEmd: Instead of using k-nearest neighbors, we also used the radius to select neighbors and the radius is a hyper-parameter.

As shown in Figure. 5(b), EQUITY2VEC achieves better performance than the above five baselines. We have the following observations: (1) Comparing with NoEmd proves the effectiveness of the information aggregated through stock embedding. (2) Comparing with PoorMan and OneEmd enables us to confirm the efficacy of learning evolving relations from k-nearest neighbors. (3) Comparing with SectEmd shows that EQUITY2VEC not just capture merely the sector information. (4) The RadiusEmbd is our variation and also achieved competitive results.



(a) *Long-short PnL.*



(b) *Long-only PnL.*

Figure 6: The cumulative PnL (Profit and Loss) curves of the top quintile portfolio. For example, on any given day, we consider a portfolio with only the top 20% strongest predictions in magnitude, against future *market excess returns*. We simulate the investment on both (a) *Long-short portfolio* and (b) *Long-index* portfolio.

Market Trading Simulation. To further evaluate our method's effectiveness, we conduct a back-testing by simulating the stock trading for three out-of-sample years. We simulate investments on our signals in two ways:(i) *Long-short* portfolio. (ii) *Long-index* portfolio: Long-only minus the market index. We conduct the trading in the daily granularity and select the stocks from the top 20% strongest forecast signals. The position of each stock is proportional to the signal (i.e., the dollar position of i-th stock is proportional to our forecast $\hat{r}_i^t$). The holding period is 5 days. [3]By allowing short-selling, we can execute on negative forecasts to understand the overall forecasting quality. Figure. 6 shows the cumulative PnL for our approach and baselines. We can see that our signals are consistently better than other baselines in both *Long-short* and *Long-index* portfolios, suggesting that our method generates stronger and more robust signals for trading.

Ablation study To test the effectiveness of different parts of the EQUITY2VEC, we conduct an ablation study that excludes components and measures the out-of-sample correlation in the year 2018. The results are shown in Figure 7. We draw two main conclusions from these results. *(i) Each component has a substantial impact on*

[3]Short is implementable in the Chinese market only under particular circumstances, e.g., through brokers in Hong Kong under special arrangements.

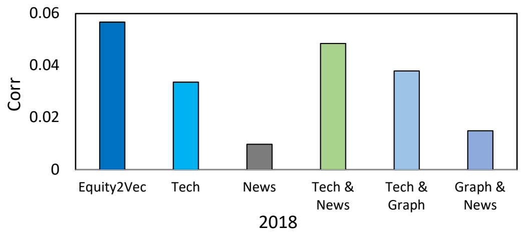


Figure 7: The performance of ablation study.

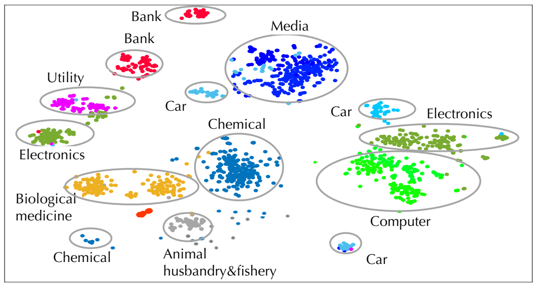


Figure 8: t-SNE of final stock representations (colors code industry sectors).

the performance: the results validate the inclusion of each component in our method. *(ii) Effect of cross-categorical learning:* the model using graph achieves higher performance than the baselines without it, thus illustrating its ability to capture and leverage latent stock interactions.

7 INTERPRETATION ANALYSIS

In this section, we assess the interpretability for stock relationships, temporal weights in LSTM, and news as follows.

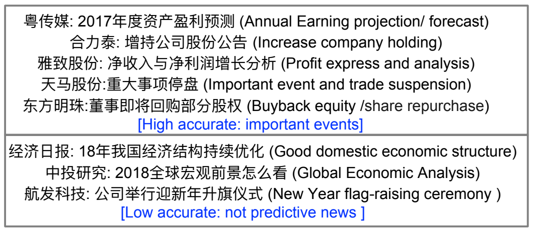
Visualizing Learned Stock Embedding. As depicted in Figure 8, we use t-SNE [37] on our final stock representation to assess the interpretation of the universe of stocks. Each dot represents a stock, and the color denotes the largest sector from Barra [13] associated with the given stock, while the text annotations represent the detailed stock categories. It shows that our Equity2Vec learns the interpretable stock representations that are well aligned with the Barra sectors.

News Interpretation. To understand the news predictive ability, we track back the news from two groups in the test dataset. We focus the samples within the smallest 5% and the samples within the largest 5% errors based on Equation 10. We extract the corresponding news and show the detailed results in Figure 9. We focus on the demonstrative news from these two groups of samples. One can see that the news in the high accuracy group contains significant events with predictive ability, while the news in the low accuracy group mainly has no apparent influence on the stock forecast.

Temporal Attention Explanation. Next, we show the overall attention weights from the testing data set in Table 2 when we use the past 5 days' historical data. The hyperparameter 5 is set by the performance in the validation dataset. The recent days have larger weights indicating the recent days play more significant roles in the prediction.

	-5day	-4day	-3day	-2day	-1day
Weights	0.0055	0.0265	0.1662	0.3064	0.4954

Table 2: The temporal overall attention weights.


粤传媒: 2017年度资产盈利预测 (Annual Earning projection/ forecast)
合力泰: 增持公司股份公告 (Increase company holding)
雅致股份: 净收入与净利润增长分析 (Profit express and analysis)
天马股份:重大事项停盘 (Important event and trade suspension)
东方明珠:董事即将回购部分股权 (Buyback equity /share repurchase)
[High accurate: important events]

经济日报: 18年我国经济结构持续优化 (Good domestic economic structure)
中投研究: 2018全球宏观前景怎么看 (Global Economic Analysis)
航发科技: 公司举行迎新年升旗仪式 (New Year flag-raising ceremony)
[Low accurate: not predictive news]

Figure 9: The demonstrative news from high accuracy and low accuracy performance.

8 RELATED WORK

Predicting equity returns (i.e., empirical asset pricing) is an extensively studied academic disciplinary that can date back to the beginning of the 20th century [6, 21], so it is impossible to provide a comprehensive review here. Instead, we focus on recent work on using machine learning to forecast asset prices.

Linear Model. Empirical asset pricing (e.g., estimating the "true price" or forecasting the future price) is an important area in Finance [3]. Linear regression has been a dominating methodology to forecast equity returns, especially for intraday tradings [5, 15, 24, 33]. Because these linear models usually use a large number of features, regularizations are usually needed [27, 29, 46, 60]. For example, recently researchers examined regularization for "ultra-high dimensional" setting, in which the number of features could be significantly larger than the number of observations [60].

Deep Learning. There are two major approaches to forecast equity returns. *Approach 1. ANN as a blackbox for standard "factors."* First, "factors" that are known to be correlated with returns are constructed. These factors can be viewed as features constructed by financial experts. Second, the factors are fed into standard ANN black boxes so that non-linear models are learned (see e.g., [22, 25] and references therein). Little effort is made to optimize ANN's architecture or algorithm. *Approach 2. Forecasting the price time-series.* This approach views the price, trading volume, and other statistics representing trading activities as time series and designs specialized deep learning models to extract signals from the time series. Little feature engineering is done for these models. See e.g., [12, 17, 35, 36, 51, 54]. Approach 1 represents the line of thought that feature engineering is critical in building machine learning models, whereas Approach 2 represents the mindset that deep learning can automatically extract features so effort on feature engineering should be avoided. The co-movement effect is often ignored by the previous studies. Only a few works on stock predictions have explored this effect [7, 16, 59]. However, [7] relies on non-public dataset and learns the relations in a static way. [16] consider only consider the relation to a particular type (sector and supply chain) and ignore the other relations, such as stocks affected by the same event. [59] has unsatisfying performance (even in their own reports on experiments) and only consider the co-movements of historical prices.

News. NLP-based techniques are developed to correlate news with the movement of stock prices. Earlier works use matrix factorization approaches (see e.g., [41, 56]) whereas more recent approaches use deep learning methods [1, 8, 11, 26]. These methods exclusively use the news to predict equity returns, and they do not consider any other "factors" that can impact the stock prices.

Factor Model (Cross-sectional Returns). The movement of two or more stocks usually can be explained by a small subset of factors. For example, Facebook and Google often co-move because their return can be explained by the technical factors. The so-called "factor model" (e.g., [14, 55, 60, 61]) can effectively capture the co-movement of prices but these methods usually rely on PCA/SVD techniques and are not computationally scalable.

Comparison. *1. Comparing to existing linear models.* Our method is more effective at extract non-linear signals. *2. Comparing to existing DL models.* We find that we need both careful feature engineering and optimizing DL techniques to use technical factors in the most effective manner, moreover, we do not restrict the stock relation into a particular type and learn the evolving relations. *3. Comparing with News/NLP-based techniques.* We do not exclusively rely on the news. Instead, we explicitly model the interaction between news and other factors so that our model avoids low quality signals (e.g., news-based signals could be essentially trading momentum), and *4. Comparing with factor models.* A key innovation of our model is the introduction of Equity2Vec component. This component models the interaction between stocks and circumvents SVD computation on large matrices.

9 CONCLUSION AND FUTURE WORK

This paper presents a novel approach to answer two research questions. *(i)* How can we interpret the relationship between stocks? *(ii)* How can we leverage heterogeneous data sources to extract high-quality forecasting power? Through extensive evaluation against the state-of-art baselines, we confirm that our method achieves superior performance. Meanwhile, the results from different trading simulators demonstrate that we can effectively monetize the signals. In addition, we interpret the stock relationships highlighting they align well with the sectors defined by commercial risk models, extract important technical factors, and explain what kind of news has more predictive power.

We identify several potential future directions. First, it is worth exploring more effective features from social media such as financial discussion forums. As individual investors often engage in insightful discussions on finance topics and stock movements, the large volume of such discussions could indicate potential upcoming major events. Second, the proposed Equity2Vec can generalized to other problems, such as mining the relations between futures.

10 ACKNOWLEDGEMENT

We thank anonymous reviewers for helpful comments and suggestions. Christopher G. Brinton was supported in part by the Charles Koch Foundation. Qiong Wu, Zheng Zhang and Zhenming Liu are supported by NSF grants NSF-2008557, NSF-1835821, and NSF-1755769. Mihai Cucuringu and Andrea Pizzoferrato acknowledge support from The Alan Turing Institute EPSRC grant EP/N510129/1. Andrea Pizzoferrato also acknowledges support from the National Group of Mathematical Physics (GNFM-INdAM), and by the EPSRC grant EP/P002625/1. The authors acknowledge William & Mary Research Computing for providing computational resources and technical support that have contributed to the results reported within this paper.

REFERENCES

[1] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. 2016. Deep learning for stock prediction using numerical and textual information. In *ICIS*.

[2] Yakov Amihud. 2002. Illiquidity and stock returns: cross-section and time-series effects. *Journal of financial markets* (2002).

[3] Turan G Bali, Robert F Engle, and Scott Murray. 2016. *Empirical asset pricing: The cross section of stock returns*. John Wiley & Sons.

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

[5] Nusret Cakici, Kalok Chan, and Kudret Topyan. 2017. Cross-sectional stock return predictability in China. *The European Journal of Finance* (2017).

[6] John Y Campbell, John J Champbell, John W Campbell, Andrew W Lo, Andrew W Lo, and A Craig MacKinlay. 1997. *The econometrics of financial markets*. princeton University press.

[7] Chi Chen, Li Zhao, Jiang Bian, et al. 2019. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In *KDD*.

[8] Raymond Chiong, Zongwen Fan, Zhongyi Hu, Marc TP Adam, Bernhard Lutz, and Dirk Neumann. 2018. A sentiment analysis-based machine learning approach for financial market prediction via news disclosures. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 278–279.

[9] Robert W Colby and Thomas A Meyers. 1988. *The encyclopedia of technical market indicators*. Dow Jones-Irwin Homewood, IL.

[10] Marcos Lopez De Prado. 2018. *Advances in financial machine learning*. John Wiley & Sons.

[11] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *IJCAI*.

[12] Jonathan Doering, Michael Fairbank, and Sheri Markose. [n.d.]. Convolutional neural networks applied to high-frequency market microstructure forecasting. In *CEEC*.

[13] Frank J Fabozzi and Harry M Markowitz. 2011. *The theory and practice of investment management: Asset Allocation, Valuation, Portfolio Construction, and Strategies*. Vol. 198. John Wiley & Sons.

[14] Eugene F Fama and Kenneth R French. 1992. The Cross-Section of Expected Stock Returns. *Journal of Finance* (1992).

[15] Eugene F Fama and Kenneth R French. 2016. Dissecting anomalies with a five-factor model. *The Review of Financial Studies* 29, 1 (2016), 69–103.

[16] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *TOIS* (2019).

[17] Guanhao Feng, Nicholas G Polson, and Jianeng Xu. 2018. Deep learning in asset pricing. *arXiv preprint* (2018).

[18] Tak-Chung Fu, Chi-Pang Chung, and Fu-Lai Chung. 2013. Adopting genetic algorithms for technical analysis and portfolio management. *Computers & Mathematics with Applications* 66, 10 (2013), 1743–1757.

[19] Ramazan Gencay and Thanasis Stengos. 1998. Moving average rules, volume and the predictability of security returns with feedforward networks. *Journal of Forecasting* (1998).

[20] Mustafa Göçken, Mehmet Özçalıcı, Aslı Boru, and Ayşe Tuğba Dosdoğru. 2016. Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications* 44 (2016), 320–331.

[21] Richard C Grinold and Ronald N Kahn. 2000. Active portfolio management. (2000).

[22] Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2018. *Empirical asset pricing via machine learning*. Technical Report.

[23] Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* (2020).

[24] Campbell R Harvey, Yan Liu, and Heqing Zhu. 2016. âĂę and the cross-section of expected returns. *The Review of Financial Studies* 29, 1 (2016), 5–68.

[25] JB Heaton, NG Polson, and Jan Hendrik Witte. 2017. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry* (2017).

[26] Ziniu Hu, Weiqing Liu, et al. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *WSDM*.

[27] Dashan Huang, Jiaen Li, and Guofu Zhou. 2019. Shrinking factor dimension: A reduced-rank approach. *Available at SSRN 3205697* (2019).

[28] Zura Kakushadze. 2016. 101 formulaic alphas. *Wilmott* (2016).

[29] Bryan T Kelly, Seth Pruitt, et al. 2019. Characteristics are covariances: A unified model of risk and return. *JFE* (2019).

[30] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[31] Charles D Kirkpatrick II and Julie A Dahlquist. 2010. *Technical analysis: the complete resource for financial market technicians*. FT press.

[32] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[33] Serhiy Kozak, Stefan Nagel, and Shrihari Santosh. 2020. Shrinking the cross-section. *Journal of Financial Economics* 135, 2 (2020), 271–292.

[34] 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).

[35] Bryan Lim, Stefan Zohren, and Stephen Roberts. 2019. Enhancing time-series momentum strategies using deep neural networks. *The Journal of Financial Data Science* 1, 4 (2019), 19–38.

[36] Tao Lin, Tian Guo, and Karl Aberer. 2017. Hybrid neural networks for learning the trend in time series. In *IJCAI*.

[37] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[38] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint* (2013).

[39] Tomas Mikolov, Kai Chen, Gregory S Corrado, and Jeffrey A Dean. 2015. Computing numeric representations of words in a high-dimensional space. US Patent.

[40] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

[41] Felix Ming, Fai Wong, Zhenming Liu, and Mung Chiang. 2014. Stock market prediction from WSJ: text mining via sparse matrix factorization. In *ICDM*.

[42] Marc-Andre Mittermayer and Gerhard F Knolmayer. 2006. Newscats: A news categorization and trading system. In *ICDM'06*. Ieee, 1002–1007.

[43] Howard Musoff and Paul Zarchan. 2009. *Fundamentals of Kalman filtering: a practical approach*. American Institute of Aeronautics and Astronautics.

[44] Christopher J Neely. 1997. Technical analysis in the foreign exchange market: a layman's guide. *Review-Federal Reserve Bank of St. Louis* 79, 5 (1997), 23.

[45] Christopher J Neely and Paul A Weller. 2011. Technical analysis in the foreign exchange market. *Federal Reserve Bank of St. Louis Working Paper No* (2011).

[46] Sahand Negahban and Martin J Wainwright. 2011. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *ANN STAT* (2011).

[47] Whitney K Newey and Kenneth D West. 1986. *A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix*. Technical Report.

[48] 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.

[49] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[50] Richard A Posner. 2014. *Economic analysis of law*.

[51] Akhter Mohiuddin Rather, Arun Agarwal, and VN Sastry. 2015. Recurrent neural network and a hybrid model for prediction of stock returns. *EXPERT SYST APPL* (2015).

[52] Hasim Sak, Andrew W Senior, and Françoise Beaufays. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. (2014).

[53] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[54] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing* 90 (2020), 106181.

[55] James H Stock and Mark Watson. 2011. Dynamic factor models. *Oxford Handbooks Online* (2011).

[56] Andrew Sun, Michael Lachanski, and Frank J Fabozzi. 2016. Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction. *International Review of Financial Analysis* 48 (2016), 272–281.

[57] Jonathan L Ticknor. 2013. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications* (2013).

[58] 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.

[59] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. 2019. AlphaStock: A Buying-Winners-and-Selling-Losers Investment Strategy using Interpretable Deep Reinforcement Attention Networks. In *KDD*.

[60] Qiong Wu, Felix M.F. Wong, Yanhua Li, Zhenming Liu, and Varun Kanade. 2020. Adaptive Reduced Rank Regression. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[61] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *KDD*.