Stock Prices Prediction of Bombay Stock Exchange using Graph Convolutional Networks

Yash R. Jain[1], Atharva D. Veer[1], Gaurav S. Sawant[1], Yash S. Jain[1], Sowmiyaraksha R. Naik[1]

[1]Yash R. Jain, Veermata Jijabai Technological Institute
[1]Atharva D. Veer, Veermata Jijabai Technological Institute
[1]Gaurav S. Sawant, Veermata Jijabai Technological Institute
[1]Yash S. Jain, Veermata Jijabai Technological Institute
[1]Sowmiyaraksha R. Naik, Veermata Jijabai Technological Institute

---***---

Abstract - *Stock prices prediction is a very hot topic of research and a lot of research is being carried out to model the stock prices for obtaining accurate predictions of the future prices of the stocks. Although the stock values are dependent on a multitude of factors; with the help of Machine Learning and due to the availability of a vast amount of stock prices data, it has been possible to predict the prices of the stocks with a good accuracy. Most of the research on stock market modeling that has been done or which is going on is done on the New York Stock Exchange (NYSE). This paper is an attempt to model the Indian Stock Market using data obtained from the Bombay Stock Exchange (BSE) with the help of Graph Convolutional Neural Networks to accurately determine the future prices of the stocks of the companies taken under consideration.*

Key Words: **GCN**: Graph Convolutional Networks, **BSE**: Bombay Stock Exchange, BSE 100 Index, **OLHC**: Open Low High Close

1.INTRODUCTION

Stocks are all of the shares into which ownership of a corporation is divided. These stocks are released by the shareholders of the company to gain funds for further growth of the organization and for earning higher profits and to create a good image of the company among the common public. A fractional ownership of the company is represented by a single share of the stock in proportion to the total number of shares of the company. If an individual holds at least one share of a company, then that individual is entitled to be a fractional owner of that company. Since the early days of the stock market, the goal of the traders and investors has been to predict the price of the stock so as to buy as low as possible and sell as high as possible, thus earning a profit. Predicting the future stock prices has always been a topic of interest for research scientists as well due to the nature of the problem. The rise or fall of any stock price is dependent upon a wide variety of factors such as company performance, number of trades taken place for that stock, demand of that stock, people's sentiments and various other factors. Thus, accurately determining the exact future prices is practically impossible. This is in accordance with the Efficient Market Hypothesis.[4] [5] However, due to the availability of a large amount of stock prices time-series data and with the emergence of Machine Learning techniques [6], it has been possible to predict the upcoming stock prices with good accuracy. By having a model that predicts with good accuracy, one can make good investment decisions and gain higher returns on a lower investment amount, thereby earning a good percentage of profits with minimal risks involved.

In the context of this project, the stock prices are being assumed to be time series data of equal intervals and we try forecasting the future time series. A time series is a sequence of observations collected over regular time intervals and is described by the change and behaviour in characteristics of a process over a period of time. If we interpret the process with the help of statistics and graphical representations, then we can model it and use it to predict the future behaviour of the stock market. Then by the help of Statistical Modelling, Deep Learning [7][8] methods we can predict the stock prices with good accuracy.

The scope of this project includes exploring the Graph Convolutional Network architectures and understanding the effect of interrelations among various companies and its impact on their corresponding stock prices. We also gather the stock data in a raw form from BSE public dataset[3], cleaning it, extracting the relevant data fields and then modelling that data into a Graph that shows the relationship between various companies and how strong or weak the relation is in form of weights.

We are focusing only upon a few of the companies listed in the Bombay Stock Exchange (BSE). BSE is an Indian Stock Exchange located in Mumbai (Bombay) and it is Asia's oldest Stock Exchange and 7th largest Stock Exchange in the World with a market capitalization of around USD 2.8 trillion as of February 2021. Yet most of the research on Stock Prices Prediction has been done on the NYSE stock market data, so we decided to carry out some extensive research on the BSE stock market data. We present a novel approach using Graph Convolution Networks to predict the Opening, Closing, Highest and Lowest price (OLHC) for a given day of stocks under consideration. There are a few papers which have performed analysis only on renowned companies,[9] We have chosen 87 companies on the basis of the BSE 100 Index and availability of data which was an important factor to be considered. We represent the company relation graph using an adjacency matrix. In order to create the adjacency matrix, we have used three different indicators[10-13] to correlate the stocks of various companies. This input along with the time series data is fed to GCN[14] architecture, so as to predict the future stock prices.

2. THEORETICAL BACKGROUND

GCNs [15] are a very powerful type of neural network architecture that are being used to do machine learning on graphs. The power of GCN is evident from the fact that even a randomly initialized two-layered GCN can provide distinctive feature representations of the nodes in the networks. Formally, a Graph Convolutional Network (GCN) is a neural network that operates on graphs. Given a graph G with V vertices and E number of edges, a GCN takes as input:

1. A $N \times F^0$ feature matrix, X, where N is the number of vertices (companies in our context) and F^0 is the number of input features (size of time series data of past several days) for each company

2. A $N \times N$ matrix representation of the graph in the form of normalized binary adjacency matrix $\bar{A}$ of G.

A hidden layer in the Graph Convolutional Network can thus be represented as $H^i = f(H^{i-1}, A))$ where $H^0 = X$, the feature matrix and f is a propagation. Each of the hidden layers H^i corresponds to an $N \times F^i$ feature matrix where each row is a feature representation of a node. These features are aggregated at each layer to form the next layer's features using the propagation rule f. Thus, the features become increasingly abstract at each consecutive layer of the GCN.

3. DATASET

3.1 Dataset description

The data used in this project is the time series data of BSE top 100 companies by market capitalization, that is the stock prices data collected at equal intervals of time (daily). The dataset consists of stock prices of various stocks over the past three years. This time series raw data was collected from quandl's BSE API [16]. For every company, the dataset contains data of daily Open, High, Low, Close, WAP, No. of shares, No. of trades, total turnover, deliverable quantity, Spread H-L, Spread C-O. The dataset was extracted from the quandl API in CSV format. The extracted raw data for the company was converted into a pandas dataframe. Date field has been converted into milliseconds since epoch. Out of these top 100 companies, 13 companies were relatively new and did not have sufficient historical data for analysis and thus were discarded. Historical data contained 656 days worth of data.

3.2 Feature Set Analysis

The Table I illustrates 4 days data for the company Reliance Industries Limited. From the table, it is evident that the number of trades is very volatile and does not have a correlation to the actual price / change in price of the given stock and thus we choose not to use the number of trades for further analysis. Similar arguments can be made for (Total Turnover, Deliverable Quantity, WAP, Number of Shares) and thus we chose to not include these set of features for the model. Furthermore, Spread H-L and Spread C-O are just spreads of High-Low and Close-Open respectively, thus using both OLHC (Open, Low, High, Close) and these spreads would be redundant and would add unnecessary complexity to the prediction model, thus we have excluded spreads for the model prediction. Thus OLHC (Open Low High Close) features of the given stock are used for prediction of the stock's new value.

3.3 Preparing Data for Model

1) Graph Generation: With the help of last 3 years (from 2018-01-01 to 2020-10-10) OLHC data for 87 companies, we generated a graph having 87 nodes (every company is represented as a node) and set of edges (relationship between companies that helps the model in prediction) [17] among these nodes.A Graph G is defined by:

- Vertices V: In our model, the vertices of the graph are the number of companies under consideration.

- Edges E: In our model, the edges represent the relationship of one company with the other company with which it shares the edge. The value of a particular edge determines how related the two companies are; higher the value, higher the relation and vice versa. We calculate the weight of each edge using different correlation techniques discussed further in the paper.

- Building Vector Representation of Company's historical data: We represent the relationship between two companies with the help of an edge having a weight proportional to the extent of relationship between them. In order to assign the weight to the edge for any pair of companies we need to create a representation of the two companies under consideration. We propose a novel approach of representing a company with the help of its historical data in a vector. Vector representation for a company represents the normalized difference between consecutive day's average stock value for that company. In the next step we compare these vectors of different companies and calculate the edge weight for all pairs of companies. Below is the formula for calculating the normalized difference between consecutive days average for a company.

$$diff_{norm} = \frac{A_{i+1} - A_i}{A_{i+1} + A_i} \quad (1)$$

where,

diff$_{\text{norm}}$ is the normalized difference between two consecutive day's average for a company, A_{i+1} is the average of the current day and A_i is the average of the previous day.

- Comparing the Vector Representation to form Graph

Once the vector representations for all the companies have been generated, we calculate the edge weight for all the pairs of companies. To calculate the edge weights we have used three methods viz. Cosine Similarity, Spearman Correlation, Pearson Correlation and take the weighted average of all the methods to calculate the final edge weight.

[13] Cosine Similarity (C_{xy}):

$$C_{xy} = \frac{x \cdot y}{|x||y|} \quad (2)$$

[12] Spearman Coefficient (S_{xy}):

$$S_{xy} = 1 - \frac{6\sum(x_i - y_i)^2}{n(n^2-1)} \quad (3)$$

[11] Pearson Coefficient (P_{xy}):

$$P_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (4)$$

Final Weight (E_{xy}):

$$E_{xy} = W_1 * C_{xy} + W_2 * S_{xy} + W_3 * P_{xy} \quad (5)$$

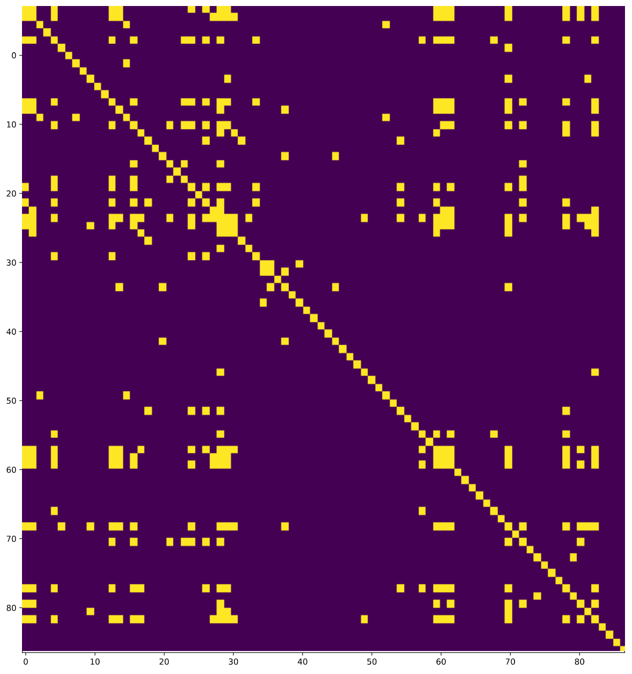

Fig. 1: Binary Adjacency Matrix

The graph is represented as an Adjacency Matrix A, where A[i][j] represents the weight of the edge connecting company i and j. Table II represents the

TABLE I: Consecutive 4 days data for Reliance Industries Ltd.

Date	Open	High	Low	Close	WAP	# Shares	# Trades	Turnover	Deli. Qty	% Deli. Qty	Spread H-L	Spread C-O
2020-10-28	2044.0	2057.9	2006.9	2010.7	2031.09	1519047	23215	3085321626	1315250	86.58	51.0	-33.3
2020-10-29	1999.4	2042.3	1990.75	2026.55	2019.52	384785	24193	777081516	61211	15.91	51.55	27.15
2020-10-30	2035.0	2064.65	2022.0	2054.35	2040.8	354434	21010	723330099	59640	16.83	42.65	19.35
2020-11-02	2033.5	2033.5	1860.0	1877.3	1925.38	1444243	99979	2780720563	454410	31.46	173.5	-156.2

TABLE II: Edges between related companies

Company A	Company B	Edge Weight
Bajaj Finance Limited	Bajaj Finserv Ltd.	0.8431
State Bank Of India	Bank Of Baroda	0.7342
JSW Steel Ltd.	Tata Steel Ltd.	0.7168
Bank of Baroda	Punjab National Bank	0.7140
ICICI Bank Ltd.	Axis Bank Ltd.	0.7695

TABLE III: Edges between unrelated companies

Company A	Company B	Edge Weight
Torrent Pharmaceuticals Ltd.	Eicher Motors Ltd.	0.0135
Colgate-Palmolive (Ind) Ltd.	Yes Bank Ltd.	-0.0193
Torrent Pharmaceuticals Ltd.	TCS Ltd.	0.0027
Infosys Ltd.	Coal India Ltd.	0.0778

highly correlated companies and that is evident in the generated graph with high edge weight. Table III represents unrelated companies and that is evident in the generated graph with edge weight being close to 0. The Graph Convolutional Network used requires a binary weighted adjacency matrix (edge weight can either be 0 (relationship does not exist) or 1 (relationship exists)). To convert our weighted graph into a binary adjacency matrix, we use the concept of thresholding, wherein we set the edge weight to 1, if it is above a Threshold value (T_H), else we set it 0. Fig. 1 represents the binary adjacency matrix (87x87) representing the relationship between all pairs of companies, In Fig. 1, a bright spot represents that there is a relation between the pair of companies and a dark spot represents no relation.

2) Processing Graph: Our final model requires an $\bar{A}$ matrix, which is a normalised adjacency matrix, which is normalised as per the degree of that particular node. For that we create a D matrix where the major diagonal represents the degree for the companies. Now, we calculate the inverse square root matrix of D and store that in $D^{-1/2}$. Finally, we get the required $\bar{A}$ matrix by performing the following matrix multiplication:

$$\bar{A} = D^{-1/2} * A * D^{-1/2} \qquad (6)$$

We used the $\bar{A}$ matrix in the prediction model.

3) Processing Time Series Data: Along with the graph $\bar{A}$, we have the stock's historical time series data. Rolling Window [18] technique is used to generate the train and test data for the model. We used the data points from the window for training and then we used that to predict the price of the first day which is not present in the window. We have chosen the window size to be 30. So, we train the model on the first 30 days of OLHC along with $\bar{A}$ and use it to predict the OLHC on the 31st day.

TABLE IV: Dimensions of test and train data for the Model

Dimensions of Time Series and Graph Train Data	
X	[600, 87, 120]
$\hat{A}$	[87, 87]
Y	[600, 87, 4]
Dimensions of Time Series and Graph Test Data	
X	[56, 87, 120]
$\hat{A}$	[87, 87]
Y	[600, 87, 4]

After combining 30 days data, we have a feature set of size 120 where we have OLHC values of 30 days and our label has the size of 4 representing OLHC of 31st day, thus from the last 3 years data, we generate a training set with 656 data points for a single company wherein every datapoint contains 120 features (OLHC of 30 days) and 4 output labels (OLHC of 31st day). Since time series data is closely linked to days/time, it cannot be split into percentages for training and testing. Thus, we consider the first consecutive 600 data points and the next consecutive 56 data points for testing. Table IV represents the dimensions of the input data for the model.

4. MODEL FORMULATION

The generated normalised graph (which is represented by Ā) is a spatio-temporal graph. A spatio-temporal graph is a graph whose structure is constant throughout all the l ayers of the entire model.

1. Inputs for the model:

    - Normalized Adjacency matrix of the graph (Ā)
    - Processed time series of historical stoc k prices (X) of every node (company) in the graph.

The normalized adjacency matrix Ā is now passed to the GCN model (Fig. 2) along with the processed time series the companies (X). This is then passed onto the GCN layers which calculates the spatial dependency from input (Ā) and temporal[19] relationship from feature time series matrix (X). The spatial dependency features calculated from the GCN layers are then passed on to the dense layers which calculate linear transformation to predict the OLHC for 87 companies. Since the data is normalized, we calculate the Root Mean Squared Error (RMSE). With the help of RMSE we calculate error for the last layer and with the help of back propagation, we calculate the gradient for all the layers. Using these gradients calculated from back propagation, Adam Optimizer updates the weights of the model and this concludes one epoch. For training of model multiple epochs are performed and our model was saturated after performing close to 3500 epochs.

2. GCN Layer:

A GCN layer takes the normalized ad- jacency matrix (Ā) and time series data (X) of compan ies as input. Then, it performs a message passing step by calculating the mean value of change of the stock price of the neighbourhood companies in the graph. We match trainable vector W with all other features on our graph and this is the convolution property of GCN. Since it is an isotropic model,the neighbours will have the same W^l. This layer outputs a vector h^{l+1} which is the input for the next hidden layer.

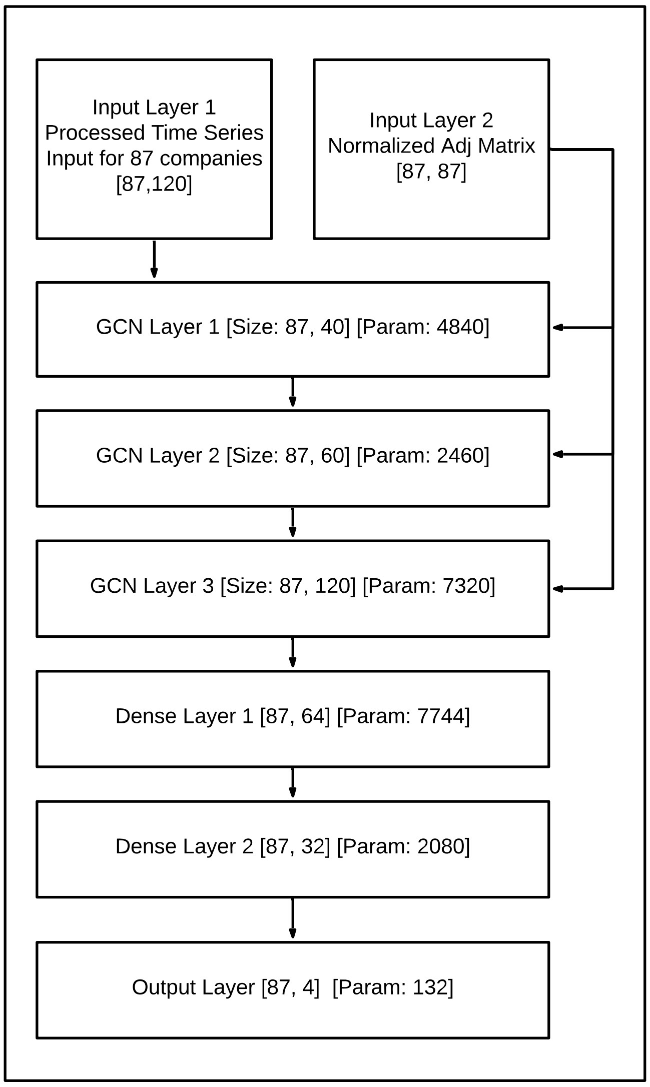
Input Layer 1
Processed Time Series
Input for 87 companies
[87,120]
Input Layer 2
Normalized Adj Matrix
[87, 87]
GCN Layer 1 [Size: 87, 40] [Param: 4840]
GCN Layer 2 [Size: 87, 60] [Param: 2460]
GCN Layer 3 [Size: 87, 120] [Param: 7320]
Dense Layer 1 [87, 64] [Param: 7744]
Dense Layer 2 [87, 32] [Param: 2080]
Output Layer [87, 4] [Param: 132]

Fig. 2: Model Architecture

Wecan stack multiple GCN layers in our model.

$$h^{l+1} = \mathrm{ReLU}(D^{-1/2} * \bar{A} * D^{-1/2} * h^{l} * W^{l}) \tag{7}$$

TABLE V: RMSEs for various windows sizes

Sr.No.	Model	Test RMSE	Train RMSE
1	GCN (window size = 3)	17.63 ±0.004	15.45 ±0.006
2	GCN (window size = 7)	17.07 ±0.017	14.83 ±0.009
3	GCN (window size = 10)	17.14 ±0.007	14.63 ±0.003
4	GCN (window size = 15)	16.87 ±0.011	14.56 ±0.008
5	GCN (window size = 30)	16.23 ±0.006	14.47 ±0.002

TABLE VI: Comparison of GCN with other models

Sr.No.	Model	Test RMSE	Train RMSE
1	GCN (window size = 30)	16.23 ±0.006	14.47 ±0.002
2	ARIMA	21.40 ±0.156	19.63 ±0.105
3	Linear Model	40.26 ±0.217	37.48 ±0.191

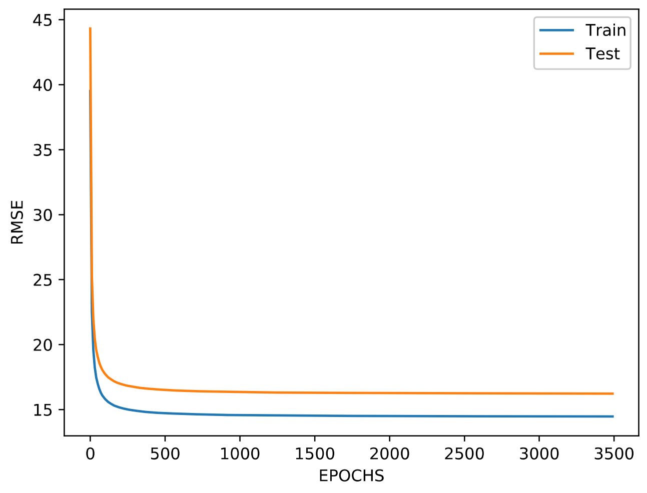

Fig. 3: Change in RMSE with epochs

5. RESULT AND ANALYSIS

Model performed best with a learning rate of 0.001. The ReLU Activation function was used for all the GCN neural network layers. Adam Optimiser was used to train the model with the loss function being Mean Squared Error. After training the model over 3500 epochs, it produced an RMSE of 16.23 on the test data and an RMSE of 14.47 on the train data. We also tested various models with different window sizes (window size represents the number of previous day's data input to the model). Table V represents the different RMSEs for various window sizes, and it is evident that the RMSE decreases with increase in the window size. Moreover we calculated metrics for various conventional models like ARIMA [20] and a Linear model. From Table VI it is clearly evident that GCN outperforms the conventional models by huge margin. This behaviour was expected because GCN is a powerful model as it considers company relationships for stock market price prediction. Overall the best performing model was GCN with a window size of 30 days. This was perhaps due to increased feature set size in comparison to other window sizes. The next best model was GCN with window size of 15 days and thus, we can empirically show that increase in window size leads to better stock prediction model.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach to model the Indian Stock Market using BSE data to predict the future prices of stocks with good accuracy by applying Graph Convolutional Networks over the time series data of past three years of BSE top 100 companies. In the paper we proved that even a per day dataset can give a good enough accuracy of predicting the future stock prices.

Paper considers BSE top 100 companies owing to complexity of project but it can be extended to include all the companies of BSE and also can be extended to include popular companies of other exchanges. Graph input Ā remains static during training, so another research opportunity could be making it dynamic to represent evolving relationships within companies with time.

REFERENCES

[1] V. H. Shah, "Machine learning techniques for stock prediction, "Foundations of Machine Learning - Spring, vol. 1, no. 1, pp. 6–12, 2007.

[2] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," inno. CONF, 2020, pp. 4541–4547.

[3] "The Bombay stock exchange (BSE)," https://www.bseindia.com/.

[4] B.G. Malkiel, "Efficient market hypothesis," in Finance. Springer, 1989, pp. 127–134.

[5] P. Patil, C.-S.M. Wu, K. Potika, and M. Orang, "Stock market prediction using ensemble of graph theory, machine learning and deep learning models," in Proceedings of the 3rd International Conference on Software Engineering and Information Management, 2020, pp.85–92.

[6] N. Milosevic, "Equity forecast: Predicting long term stock price movement using machine learning," arXivpreprint arXiv:1603.00751, 2016.

[7] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwanaet al., "Deep learning for stock market prediction,"Entropy, vol. 22, no. 8, p. 840, 2020.

[8] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in Twenty-fourth international joint conference on artificial intelligence,2015.

[9] Y. Liu, Q. Zeng, J. Ordieres Mer´e, and H. Yang,"Anti cipating stock market of the renowned companies:A knowledge graph approach,"Complexity, v ol. 2019,2019.

[10] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficient on the same sets of data," 2011.

[11] L. Myers and M. J. Sirois, "Spearman correlation coeffi-cients, differences between,"Encyclopedia of statistical sciences, 2004.

[12] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson Correlation coefficient," inNoise reduction in speech processing. Springer, 2009, pp. 1–4.

[13] S. Van Dongen and A. J. Enright, "Metric distances derived from cosine similarity and pearson and spearman correlations," arXiv preprint arXiv:1208.3145, 2012.

[14] Y. Chen, Z. Wei, and X. Huang, "Incorporating cor-poration relationship via graph convolutional neural networks for stock price prediction," inProceedings of the 27th ACM International Conference on Information And Knowledge Management, 2018, pp. 1655–1658.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks,"arXiv preprintarXiv:1609.02907, 2016.

[16] "Quandl api," https://www.quandl.com/.

[17] J. Ye, J. Zhao, K. Ye, and C. Xu, "Multi-graph convolu-tional network for relationship-driven stock movement prediction,"arXiv preprint arXiv:2005.04955, 2020.

[18] D. Matsunaga, T. Suzumura, and T. Takahash, "Ex ploring graph neural networks for stock market predic tions with rolling window analysis,"arXiv preprintarXiv: 1909.10660, 2019.

[19] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.S. Chua, "T emporal relational ranking for stock prediction,"ACM T ransactions on Information Systems(TOIS), vol. 37, no. 2, pp. 1–30, 2019.

[20] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock Price prediction using the arima model," in2014UKSimAMSS 16th International Conference on Computer Modelling and Simulation. IEEE, 2014, pp. 106–112.