Harnessing Convolution and PageRank in Graph-Based Models for Enhanced Stock Selection Amid Market Volatility

Siu Tung Wong[1], Pin Ni[2][a], Francesca Medda[2][b], and Victor Chang[3][c]

[1]*Institute of Mathematical, Engineering and Physical Science, King's College London, London, UK*

[2]*Institute of Finance and Technology, University College London, London, UK*

[3]*Department of Operations and Information Management, Aston Business School, Aston University, Birmingham, UK*

siu.wong@kcl.ac.uk;{pin.ni.21, f.medda}@ucl.ac.uk; v.chang1@aston.ac.uk

Keywords: Adjacency Matrix, Convolution, Graph Networks, Stock-Picking, Volatility

Abstract: In a financial landscape marked by the aftermath of the pandemic and the inherent volatility of today's markets, investors and fund managers grapple with the challenge of selecting stocks that can withstand short-term fluctuations and yield long-term gains. Traditional strategies, deeply rooted in a company's financial metrics, confront the capricious nature of contemporary markets that are heavily influenced by a myriad of fast-changing factors. Recognizing the need for tools that can navigate through the confluence of these complex conditions, this paper introduces a graph-based approach. Employing a single-layered convolution method applied to a curated list of stocks, the study focuses on crafting a Graph-and-PageRank-based ranking system that directs investors to the most promising stocks. The approach leverages volatility and relative market capitalization as pivotal factors to gauge the interconnected influence of stocks, embodying the multifaceted decision-making process akin to that of a seasoned investor. This methodology not only aids in mitigating risks in the short term but also sets the stage for potential long-term investment strategies, paving the way for robust financial modeling that is both responsive to market dynamics and intuitive to the investor's stock-picking rationale.

1 INTRODUCTION

Investors and fund managers have access to a variety of techniques and mathematical tools for selecting profitable stocks. However, many of these methods are biased towards current market conditions, which can pose challenges during uncertain times, such as the aftermath of the 2019 pandemic. Even fundamental traders, who traditionally rely on analyzing a company's financial health and market position, have faced losses due to the unpredictability of today's markets (Ülkü et al., 2023). It's important to recognize that while fundamental traders may experience short-term losses, they are often positioned to succeed in the long run (Lento and Gradojevic, 2022). Their strategies are grounded in historical data and models, which tend to hold true over extended periods. However, success in the long term often requires having sufficient capital to weather ongoing losses for several years while waiting for market conditions to align with their strategies.

Therefore, during the short term, traders and investors should have a mechanism that would classify the risks their investments presented. Currently, many neural network-based models, such as Recurrent Neural Networks (RNN), Deep Reinforcement Learning (DRL) and Graph Neural Networks (GNN), are present. Still, only some are explicitly targeted to financial modelling (Wu et al., 2023; Ni et al., 2022; Li et al., 2020; Li et al., 2019). Thus, there is a need to develop a comprehensive model utilising certain concepts presented in GNN, such as the convolution between nodes, to present a dynamic representation of a complex financial system; this process imitates how a trader selects certain stocks based on various external pieces of information, such as news, personnel, inter-company relations, price, volatility, ..., thus the paper presents a relative risk assessment between stocks, through the price movement and their interactions.

Our work can be divided into the following steps:

- Create a comprehensive system for ranking entities based on their level of volatility, utilising a

[a] https://orcid.org/0000-0003-4516-1249
[b] https://orcid.org/0000-0002-7951-7680
[c] https://orcid.org/0000-0002-8012-5852

Normalised Volatility Ranking Metric. For example, stocks with high levels of volatility would be ranked lower, while those with stable prices would receive a higher ranking. Additionally, there are further metrics regarding the "social networks" between interrelated companies and various fundamental financial data, alternative data sources, and market indicators, which could all be grouped as an additional metric.

- Create a series of interlinked nodes to form a "complete graph" - alternatively, an incomplete graph may be defined. Still, it must remain connected to maintain its rank, as the rank of a disconnected section is not well-defined to the other interconnected components. This graph should have a spatial dimension with a minimum dimensionality of one. The nodes within this graph can represent stocks, social networks, or companies' environmental impact scores.
- By utilising the abovementioned metrics, we can define a "signal", which would be sent out to neighbouring nodes, such that each node would be influenced by the state of activity of their surrounding nodes, resulting in a convolution; modelling of the convolution would describe how this signal's propagation would alter the connected nodes. For instance, if there is unfavourable news about Tesla's electric vehicle sales, it may negatively affect the entire electric vehicle market. As a result, this signal would lead to a general decrease in the investment potential of these stocks.
- To improve the visual representation of data on a graph by incorporating a ranking mechanism. This mechanism transforms the space of an NxN matrix to a more compact Nx1 vector, where each element represents the corresponding ranking of nodes. By utilising this approach, the user can quickly identify the most significant nodes in the graph based on their ranking, leading to a clearer understanding of the underlying data.

Our code is available here: https://github.com/UCL-IFT/D-TMM

2 RELATED WORK

Scarselli et al. (Scarselli et al., 2008) describe how traditional neural network methods adapt to accommodate data encapsulated in graph structures. GNNs can be used to process various graph types, such as acyclic, cyclic, directed, and undirected graphs; through the function, $f : G \times v \to \mathbb{R}^n$, it is possible to map a graph and one of its nodes into an n-dimensional Euclidean space, thus encoding the information from graphs and nodes into vectors; this mapping allows for the representation of graph-structured data by incorporating node relationships.

Saha et al. (Saha et al., 2021) shed light on the complexities of predicting stock price movements through innovative methods. It discusses the challenges of predicting stock price movements and the influence of stock relations on prediction tasks. The introduction highlights the limitations of existing evaluation measures and proposes a new metric, normalized rank biased overlap (NRBO@k), for stock ranking prediction. Additionally, the document outlines the use of graph-based approaches and node embedding techniques for improved stock ranking prediction performance.

Sarmah et al. (Sarmah et al., 2022) employ the Node2Vec algorithm to unravel the complexities of stock interconnections reflected in correlation matrices. Node2vec compresses the network into a lower dimensional space and evaluates the embedding.

Ma et al. (Ma et al., 2024) delve into stock ranking for portfolio optimisation through the use of a multi-task learning model with a heterogeneous graph attention network (HGA-MT). HGA-MT optimises its portfolio by integrating risk and identifying stock relationships, resulting in improved ranking accuracy and investment profits.

3 METHODOLOGY

3.1 Normalised Volatility Ranking Metric

The Normalised Volatility Ranking Metric (NVRM), would sort the given set of data and provide it a rank that represents the level of volatility relative to the initial set

$$\{A\} : O \to B, \text{ s.t. } \{A\} \in \{S\} = \{\{\alpha_1\}, \{\alpha_2\}, \dots\}, \tag{1}$$

where $\{S\}$ is the set of initial data provided, O is some operation applied to each of the element of $\{S\}$ represented as $\{A\}$ and B the resultant real value can be presented as $B \in \{S'\}$ but, $B \notin \{S\}$. Thus the above statement shows a disconnection between the input and output data ($\{A\}$ and B respectively), thus showing a lack of representation for B, in this report, B will be the value of volatility for every $\{A\}$.

Now define the operator O which would return the volatility. Firstly the set A can be defined as

$$\{A\} = \{(X)_{t_0}, (X)_{t_1}, (X)_{t_2}, \ldots, (X)_{t_{now}}\}, \quad (2)$$

with $(X)_{t_i}$ as the value at a time t_i. The gradient at every point relative to a specific point in time is,

$$\left\{ \left(\frac{(X)_{t_1} - (X)_{t_0}}{t_1 - t_0} \right), \left(\frac{(X)_{t_2} - (X)_{t_1}}{t_2 - t_1} \right), \ldots, \left(\frac{(X)_{t_{\text{now}}} - (X)_{t_{\text{now}-1}}}{t_{\text{now}} - t_{\text{now}-1}} \right) \right\}. \quad (3)$$

Let the gradient of the set $A = \Delta A$ where

$$\Delta A = \{(X)_{t_1 - t_0}, (X)_{t_2 - t_1}, \ldots, (X)_{t_N - t_{N-1}}\}; \text{now} \equiv N. \quad (4)$$

Apply the sigmoid function (σ) onto ΔA such that

$$\sigma\{(X)_{t_1 - t_0}, \ldots, (X)_{t_N - t_{N-1}}\} \in \mathbb{R}^+, \quad (5)$$

the volatility can be represented as

$$\sum_{i=0}^{N-1} \sigma(X)_{t_{i+1} - t_i} = \text{V}. \quad (6)$$

3.2 Defining the Graph

For a general assumption, nodes that can be classified as

$$n_1 \in \{S\}; n_2 \in \{S\}, \quad (7)$$

then n_1 and n_2 are interconnected nodes. The adjacency matrix can be represented as

$$A_{ij} = \begin{pmatrix} 0 & A_{12} & A_{13} & \cdots & A_{1N} \\ A_{21} & 0 & A_{23} & \cdots & A_{2N} \\ A_{31} & A_{32} & 0 & \cdots & A_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & A_{N3} & \cdots & 0 \end{pmatrix}, \quad (8)$$

where the elements of A_{ij} would have a value of $1 - \delta_{ij}$, and A_{ij} denotes node from $j \to i$. The weight of the nodes can be estimated by looking at the importance of j relative to i (such as a relative difference between market caps in the case of analysing stocks)

$$\varepsilon_{ij} A_{ij} \text{ for } \varepsilon_{ij} \in (0, 1]. \quad (9)$$

3.3 Effects of NVRM on Graphs

The weighted adjacency matrix

$$A^*_{ij} = \varepsilon_{ij} A_{ij}, \quad (10)$$

can be transferred through the operator V by representing the operator as a matrix

$$V_{jj} = \begin{pmatrix} V_1 & 0 & 0 & \cdots & 0 \\ 0 & V_2 & 0 & \cdots & 0 \\ 0 & 0 & V_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & V_N \end{pmatrix}, V_j = \sum_{i=0}^{N-1} \sigma(X_j)_{t_{i+1} - t_i}. \quad (11)$$

Now the adjacency matrix would have each of its elements affected by a corresponding V_{jj} where

$$A^*_{ij} V^{-1}_{jj} = \varepsilon_{ij} A_{ij} V^{-1}_{jj}, \quad (12)$$

where for A'_{21} representing node $1 \to 2$, would have V_{11}, thus depending on the value of V_{11}, the weight of node 1's signal to node $2, 3, \ldots, N$ would be improved or reduced.

3.4 Combining NVRM with Factors

NVRM can be represented as a diagonal matrix shown in section 3.3; thus combining its results with other factors is relatively simple as long as whatever factors perverse the properties of the matrix

$$V_{jj} : F \to V^*_{jj}, \quad (13)$$

where the factors F would usually take the form of a diagonal matrix with elements $F_{jj} \in (0, 1]$, and the operation between V_{jj} and F_{jj} can be represented as

$$V_{jj} \times F_{jj} = V^*_{jj}. \quad (14)$$

3.5 Signal Processing and Ranking Result

The "signal", represented by the resultant coefficient of each element in the matrix

$$A_{ij}(V^*_{jj})^{-1} = \begin{pmatrix} 0 & A^*_{12}(V^*_2)^{-1} & \cdots \\ A^*_{21}(V^*_1)^{-1} & 0 & \cdots \\ A^*_{31}(V^*_1)^{-1} & A^*_{32}(V^*_2)^{-1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \quad A^*_{ij} = \varepsilon_{ij} A_{ij}, \quad V^*_{jj} = V_{ij} F_{jj}. \quad (15)$$

Select a specific node to act as the receiver of the signal. For example, let node one be the receiver:

$$A^*_{i1}(V^*_{11})^{-1} \to A^*_{i1}(V^*_{11})^{-1} + \frac{1}{N} \sum_{j=1}^{N} A^*_{1j} V^*_{jj} (1 - \delta_{1j}) = \alpha_{i_1}, \quad (16)$$

represents the influence of all other nodes to node one, such that if V^*_{jj} is larger than $\sum A^*V^*(1-\delta)/N$ would be large, thus putting more weight onto node 1.

Alpha (α_{i_1}) is the value of node one relative to other nodes after the processing of the "signal". (α_{i_1}) would go into a set such that it would not influence the weight of the signal when node one transfers to other nodes

$$A^*_{i1}(V^*_{11})^{-1} \in \{S\}, \alpha_{i_1} \in \{S^*\}. \tag{17}$$

The physical significance of (α_{i_1}) represents: even though node one might have a low $A^*_{i1}(V^*_{11})^{-1}$ value (potentially due to a high value of NVRM or additional factors), but though observing other connected nodes, the relative value of node one could have risen within this set (this process resembles a graph convolution network), however for a fully connected graph, would be lightly impacted, but one can easily define a non fully connected network in this scenario.

After the signal processing is completed for each node, reform the matrix:

$$\alpha_{ij} = \begin{pmatrix} 0 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1N} \\ \alpha_{21} & 0 & \alpha_{23} & \cdots & \alpha_{2N} \\ \alpha_{31} & \alpha_{32} & 0 & \cdots & \alpha_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{N1} & \alpha_{N2} & \alpha_{N3} & \cdots & 0 \end{pmatrix}, \tag{18}$$

Now perform a random walk process, using the PageRank method to obtain the final ranking result.

$$r_j = \sum_{i=1}^{N} \alpha_{ij} r_i \Rightarrow r^* = \alpha r\,; r = \begin{pmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \end{pmatrix}, \tag{19}$$

The rank of node one (r_j) is represented by the sum of surrounding nodes $\alpha_{ij} r_i$. Applying the iterative formula, until the result of $r^* \approx r$.

$$r^{k+1} = \alpha r^k \rightarrow r = \alpha r, \tag{20}$$

Taking the inverse of r and raking it would provide the most influential node to the least influential node.

4 EXPERIMENTS

The dataset used in this study is displayed in Figure 4 of the Appendix. It comprises six stocks from the automotive industry, namely Tesla, the biggest electric automotive producer, and five general automotive producers who manufacture electric and non-electric vehicles. The dataset is chosen to serve two primary purposes. Firstly, to compare the potential relative risk associated with the list of automotive companies, and secondly, to examine the impact of a (relatively) purely electric vehicle compared to a general vehicle company.

Define the volatility of the list of stocks through Algorithm 1, where the result is a list of positive valued elements whose position is relative to the order of inputted stocks.

Algorithm 1: Stock Volatility Calculation

Input: Price_List_High, Price_List_Low
Output: Stock Volatility

for *i from 1 to n* **do**
  $\zeta_i(a_i\,, b_i) \equiv a_i(t_i) - b_i(t_i)$;
  $a_i(t_i), b_i(t_i) \in$ *Price_List_High*, *Price_List_Low*;
end
for *i from 1 to n-1* **do**
  $V(\zeta_i) = |\zeta_{i+1} - \zeta_i|$;
  $StockVolatility.append(V(\zeta_i))$;
end
return $\sum_{i=1}^{n-1} |V(\zeta_i)|$

Define the adjacency matrix for the list of stocks, assuming each of those stocks is connected.

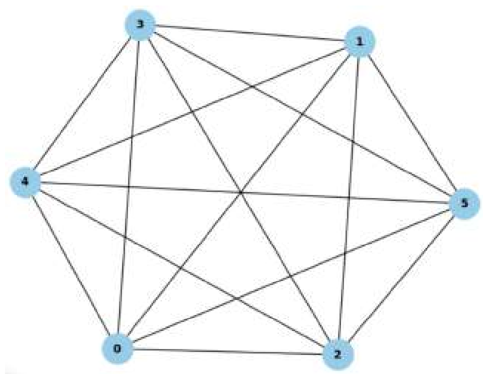

Figure 1: The non-directed fully connected adjacency matrix graphs representing the six chosen stocks

Incorporate the volatility factor into a diagonal matrix. When applied to the adjacency matrix, the resulting matrix would contain an associated volatility parameter for each element. In addition, the weight of interaction between stocks would depend on their market cap. As a result, each element of the adjacency matrix would be modified based on its relative market cap with its neighbouring market caps. These two factors combined would produce a new adjacency matrix that incorporates both volatility and market cap information into its structure:

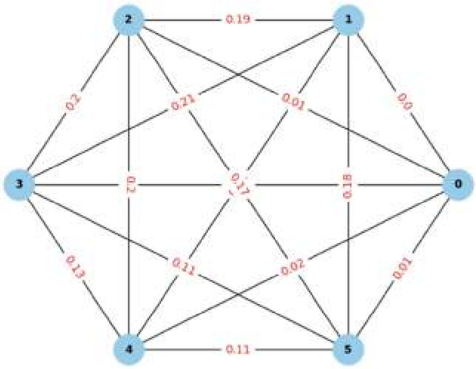

Figure 2: After transforming the adjacency matrix with the market cap coefficients and the volatility matrix.

Defining a single convolution layer such that the neighbours of a particular node's weight and volatility would affect the node's value (This process is represented in section 3.5) shown in Algorithm 2; the result of this is a matrix where each element has been convoluted with its neighbours. Passing the convoluted matrix into a random walk algorithm would result in ranking the most to least important stock; this process considers a stock whose volatility is low, the market cap is high and is surrounded by similar stocks to be ranked highly, and the inverse is true.

Algorithm 2: Convolution Matrix Calculation

Input: matrix
Output: convolution_matrix

convolution_matrix = zeros_like(matrix);
rows, cols = shape(matrix);
for *i* ***from*** *1* ***to*** *rows* **do**
  for *j* ***from*** *1* ***to*** *cols* **do**
    current_value = matrix[i, j];
    sum_neighbors = 0;
    for *m* ***from*** *max(0, i - 1)* ***to*** *min(rows, i + 2)* **do**
      for *n* ***from*** *max(0, j - 1)* ***to*** *min(cols, j + 2)* **do**
        if $(m \neq i$ **or** $n \neq j)$ ***and*** $(m \neq i$ **or** $n \neq j)$ **then**
          sum_neighbors += matrix[m, n];
        end
      end
    end
    convolution_matrix[i, j] = sum_neighbors;
  end
end
fill_diagonal(convolution_matrix, 0);
return *convolution_matrix*

Stock	Value
TSLA	$8.58540568 \times 10^{-23}$
TM	$3.93727503 \times 10^{-23}$
BMWYY	$1.13587812 \times 10^{-23}$
VWAGY	$1.11847805 \times 10^{-23}$
F	$7.84716167 \times 10^{-24}$
GM	$6.95398941 \times 10^{-24}$

Table 1: Ranking result for stocks from most desirable to least desirable.

5 RESULTS AND ANALYSIS

The results from section 4 for the list of stocks: TSLA (Tesla), GM (General Motors), F (Ford Motor), BMWYY (Bayerische Motoren Werke AG (BMW)), VWAGY (Volkswagen AG), and TM (Toyota Motor) are shown as follows:

$$\text{RandomWalk Scores}: \begin{bmatrix} 8.58540568 \times 10^{-23} \\ 6.95398941 \times 10^{-24} \\ 7.84716167 \times 10^{-24} \\ 1.13587812 \times 10^{-23} \\ 1.11847805 \times 10^{-23} \\ 3.93727503 \times 10^{-23} \end{bmatrix}$$

Where the values of the list of stocks are, the order from most to least desirable is shown as: PageRank Scores (ordered from highest to lowest):

5.1 Analysis from Volatility and Market Cap

By analysing the initial volatility and market cap before applying a convolution, we can judge whether or not the results produced are physically relevant. Based on the observation from Figure 5 in the Appendix, the volatility of these six stocks can be ranked as follows: Based on these two results, which stock would be ranked higher than the rest needs to be clarified. However, looking at the ratio of market cap to volatility, we can infer that TM would be the highest at 39.64, next would be TSLA at 11.97, then BMWYY at 9.67, VWAGY at 9.51, F at 8.52 and GM at 8.05; thus an assumption that TM should take the first spot followed by TSLA is a reasonable inference; however, the predicted results shows that TSLA overtook TM.

Nevertheless, the results are within a stock trader's expected range with external information; applying

Stock	Value	
	Price	Market Cap
TSLA	51.82	620.302
TM	36.99	308.331
GM	27.53	45.588
VWAGY	7.78	73.941
BMWYY	7.76	75.042
F	5.66	48.24

Table 2: Stock Prices and Market Caps

traditional technical analysis, such as observing the trends, volume, and shapes of the graph, an investor would come to a similar consensus (Murphy and Murphy, 1999) from purely using the information provided in Figure 6 of the Appendix. Therefore, the current algorithm could mimic how a realistic person would pick stocks while leaving the possible influence of external information as additional factors to be included.

5.2 Analysis from Graphs and Convolution of Nodes

We look at the network of these six stocks to understand why our inference could have been more accurate.

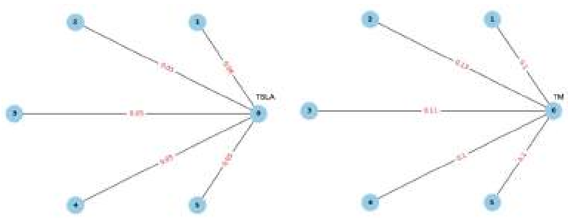

Figure 3: The left image shows how all other nodes interact with TSLA, while the right image shows the interaction of all other stocks interacting with TM.

From Figure 3, TM is shown to receive a larger signal than TSLA from its surrounding nodes. Therefore, its net value should be higher. Therefore, the result must be due to the ranking algorithm. From C_{ij} in the Appendix, a particular path may exist for this configuration such that a random walk would continuously loop back to TSLA.

5.3 Results from Different Lists

The effects of removing stocks from the lists show a drastic but predictable change to the overall results. Removing TM provided the following result:

Next are the results of the algorithm but with GM removed:

Within this list of stocks, the actual value does

Stock	Value
TSLA	$1.41230895 \times 10^{-8}$
GM	$1.08556610 \times 10^{-9}$
F	$1.24268668 \times 10^{-9}$
BMWYY	$1.82158378 \times 10^{-9}$
VWAGY	$1.81979383 \times 10^{-9}$

Table 3: Ranking result for stocks from most desirable to least desirable with TM removed.

Stock	Value
TSLA	6.91688607
F	0.87162314
BMWYY	1.20399387
VWAGY	1.20028869
TM	3.67582485

Table 4: Ranking result for stocks from most desirable to least desirable with GM removed.

not necessarily matter; within their own set, only the ranking at the end matters. However, this algorithm can be easily expanded to include two disjointed graphs. In contrast, in this case, the values would matter as they could represent the overall financial stability of a group of stocks relative to another group.

The individual results from Tables 1, 3 and 4 can be used as part of the decision-making process; as a highly ranked stock would need to have both stability, a relatively high market cap and be surrounded by relatively stable stocks. Thus an investor with a finite amount of capital and a particular interest in a specific sector can use the algorithm in the paper to obtain a result for an N number of related stocks, and the investor can focus on prioritising further research on only the highly ranked stocks.

6 CONCLUSION AND FUTURE WORK

This paper presents an alternative approach to analysing the relative risk for a list of stocks by applying factors to determine the weight each stock holds (where the weight is a combination of the stock's volatility and relative market-cap) than through the process of a single layer convolution; a stock would be influenced by its neighbour's weight and through the process of a random walk, a list of result where the most favourable stock would be ranked highest and unfavourable stock the lowest. The result for the six selected automotive stocks, TSLA, GM, F, BMWYY, VWAGY and TM, is within the range of expectation of what a stock trader would rank without the influence of external information. Therefore, the algorithm successfully mimics how a trader would rank particular stocks while retaining the possibility of external information by incorporating more factors.

However, due to the limited information available, only two factors were selected: volatility and market cap; while those are important factors, they might not necessarily be what influence the movement of a stock's price; the sentiment of the investors is much more important; where they would consider, the environmental impact, government regulations, the balance sheet, and many other factors of companies before deciding to invest. Additionally, the operation where factors are multiplied with each other might not be accurate as it gives equal weights to every factor; thus, to more factors, each factor should have its respective weight. Finally, the convolution algorithm currently takes only a single value from surrounding nodes; it could improve the algorithm if a tensor could be defined where each node is comprised of multiple values instead of a matrix. Thus a direct focus should be developing a robust system to represent the investor's sentiment of a particular investment.

7 ACKNOWLEDGEMENTS

This research is partly supported by VC Research (VCR 0000230) for Prof Chang.

REFERENCES

Lento, C. and Gradojevic, N. (2022). The profitability of technical analysis during the covid-19 market meltdown. *Journal of Risk and Financial Management*, 15(5).

Li, Y., Ni, P., and Chang, V. (2019). An empirical research on the investment strategy of stock market based on deep reinforcement learning model. In *Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS)*, volume 1, pages 52–58. SciTePress.

Li, Y., Ni, P., and Chang, V. (2020). Application of deep reinforcement learning in stock trading strategies and stock forecasting. *Computing*, 102(6):1305–1322.

Ma, Y., Mao, R., Lin, Q., Wu, P., and Cambria, E. (2024). Quantitative stock portfolio optimization by multi-task learning risk and return. *Information Fusion*, 104:102165.

Murphy, J. J. and Murphy, J. J. (1999). *Technical analysis of the financial markets*. New York Institute of Finance, Fishkill, N.Y.

Ni, P., Yuan, Q., Khraishi, R., Okhrati, R., Lipani, A., and Medda, F. (2022). Eigenvector-based graph neural network embeddings and trust rating prediction in bitcoin networks. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 27–35.

Saha, S., Gao, J., and Gerlach, R. (2021). Stock ranking prediction using list-wise approach and node embedding technique. *IEEE Access*, 9:88981–88996.

Sarmah, B., Nair, N., Mehta, D., and Pasquali, S. (2022). Learning embedded representation of the stock correlation matrix using graph machine learning.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Wu, S., Irsoy, O., Lu, S., Dabravolski, V., Dredze, M., Gehrmann, S., Kambadur, P., Rosenberg, D., and Mann, G. (2023). Bloomberggpt: A large language model for finance.

Ülkü, N., Ali, F., Saydumarov, S., and İkizlerli, D. (2023). Covid caused a negative bubble. who profited? who lost? how stock markets changed? *Pacific-Basin Finance Journal*, 79:102044.

APPENDIX

Figure 4: The list of data, obtained from Yahoo Finance, used for the 6 stocks, number from 0 to 5 and they are respectively (names are given in their stock codes), TSLA, GM, F, BMWYY, VWAGY, TM. The data are comprised of the daily high and low values of the six stocks from February 22 to April 22 in the year 2020, this data range holds no specific significance as it was chosen randomly.

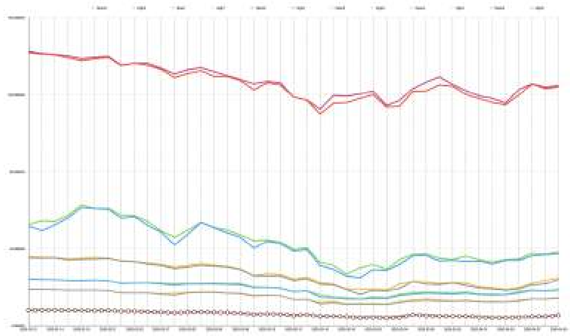

Figure 5: The line graph plotted using data from Figure 4, the y-axis represents the trading price, and the x-axis is the time-step.

Figure 6: A closer look at the six particular stocks, the difference in volatility and area between the daily high and low, along with the specific trend that appeared between March 13 to 24 can be used for technical analysis.

$$C_{ij} = \begin{bmatrix} 0 & 0.05809312 & 0.04702267 \\ 0.04517292 & 0.04543722 & 0.04732483 \end{bmatrix}$$

$$\begin{bmatrix} 0.79045543 & 0 & 0.86607519 \\ 0.84090632 & 0.84450247 & 0.87018659 \end{bmatrix}$$

$$\begin{bmatrix} 0.60464878 & 0.81846261 & 0 \\ 0.65232612 & 0.65572458 & 0.67999671 \end{bmatrix}$$

$$\begin{bmatrix} 0.37340235 & 0.51085043 & 0.41934134 \\ 0 & 0.40623591 & 0.42183901 \end{bmatrix}$$

$$\begin{bmatrix} 0.38117958 & 0.5206743 & 0.42780262 \\ 0.41228486 & 0 & 0.43033748 \end{bmatrix}$$

$$\begin{bmatrix} 0.09520835 & 0.12866065 & 0.10638905 \\ 0.10266772 & 0.10319943 & 0 \end{bmatrix}$$