1 Introduction

Time series forecasting plays a vital role in a wide range of real-world applications, from economics and finance to transportation. Predicting future trends based on historical data can help us make better decisions. For example, if we can get accurate forecasts of the stock and the commodity markets, we can make considerable profits from the forecasts. Moreover, if we can forecast the trend of COVID-19 new cases, we can allocate medical resources to prevent the epidemic outbreak in advance.

Multivariate time series data contains time series from multiple interlinked data. In addition to forecasting the trend based on historical temporal patterns within each time series, we can utilize the correlations between different time series to improve the accuracy of forecasting. For instance, as shown in Figure 1, the prices of crude oil and gasoline are highly correlated and the rise in crude oil may indicate a rise in the price of gasoline in the near future. The observation motivates that utilising the information from both time series would help us better forecast the trend of each one.

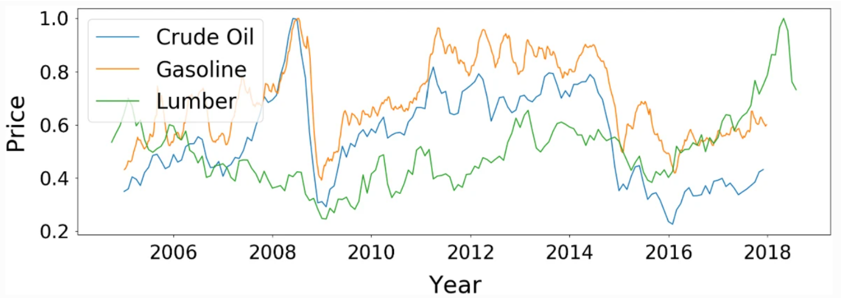

Figure 1: Historical prices of crude oil, gasoline and lumber

Significance and Novelty Making accurate multivariate time series forecasting is challenging, especially in some complicated scenarios such as the stock market, where the correlations between different stocks are not fixed and difficult to capture. Some traditional statistical methods may fail because they assume there are linear correlations among time series[1]. Besides, deep learning models involving LSTM [2] and GRU[3] do not perform well because it is difficult for them to extract spatial and temporal features jointly[4]. Aiming to capture inter-series correlation and intra-series patterns jointly, we proposed a GNN-GLU model to shuttle between the temporal and spectral domains and make forecasts based on information from multivariate time series data. To verify the significance of our model, we also conduct experiments on stock forecasting and COVID-19 new case forecasting with real-world data.

2 Methodology

2.1 Overview

The illustration of our model is shown in Figure 2. Given multivariate time-series $\mathbf{X}$ as input, we first generate the correlations among different time series and build the graph $\mathcal{G}$. Then we put $\mathcal{G}$ into the following three layers. Firstly, the **Spectral Layer** contains a Graph Fourier Transform, an Intra-series Layer, a Spectral Graph Convolution, and an inverse Graph Fourier Transform. It mainly captures the inter-series relationship. Secondly, the **Intra-series Layer** performs Discrete Fourier Transform, 1D Convolution, GLU, and inverse Discrete Fourier Transform. It mainly learns the intra-series features. Thirdly, the **Temporal Layer** applies the GLU and Fully-connected layer (FC), making the prediction. The loss function of the GLU-GNN forecasting network can be written as:

$$\mathcal{L}\left(\widehat{\boldsymbol{X}}_{\boldsymbol{t}}, \boldsymbol{X}_{\boldsymbol{t}}; \boldsymbol{\Theta}\right) = \sum_{t=K}^{T} \left\| \widehat{\boldsymbol{X}}_{\boldsymbol{t}} - \boldsymbol{X}_{\boldsymbol{t}} \right\|_2^2 + \frac{\lambda}{2} \|\boldsymbol{\Theta}\|_2^2 \tag{1}$$

where the $\hat{X}_t$ is the forecast value for the timestamp t, the X_t is the ground truth value of the timestamp t, the K is the sliding window size of the history data, and the Θ denotes all parameters in the network.

2.2 Graph Formulation

Multivariate time series can be denoted as a tensor $X \in \mathbb{R}^{N \times T}$, where N is the number of time series and T is the number of time slots. For each time t, each node i consists of a sequence of previous K values $(X_{i,t-K}, \cdots, X_{i,t})$ for time series S_i. An adjacency matrix $W \in \mathbb{R}^{N \times N}$ to reflect the correlations among different time series, where the entry w_{ij} represents the weight of the edge between node i, j.

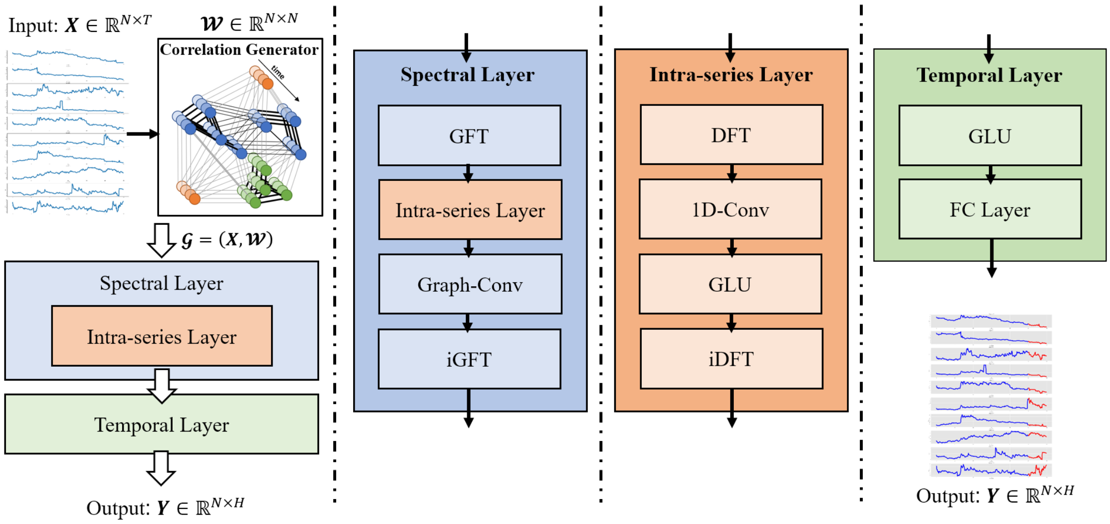

Figure 2: Overall architecture of GNN-GLU model

2.3 Correlation Generation

In order to derive an adjacency matrix W that well reflects the correlations among time series in different scenarios, we propose three approaches to achieve it.

- **Human knowledge** We can design the matrix by the knowledge towards different time series, for example, in the stock forecasting task, we can formulate the matrix based on some factors of the stock market or the commodity market[5].
- **Physical meaning in real world** There is some useful information in the physical world we can utilize, for example, we can use the distances between different provinces in China to reflect their geometric relationships, which are stationary. Besides, we can utilize the population migration data which are dynamic.
- **Automatic generation based on self-attention mechanism** Note that the matrices formulated from the above two methods are predefined by ourselves, there is another way to automatically generate the matrix based on the self-attention mechanism[6]. First, we feed the sequence of previous k values $(X_{t-K}, \cdots, X_t)$ into a Gated Recurrent Unit (GRU) layer to get the hidden state of each timestamp and represent each time series (node) as its last hidden state H_t. Finally, we calculate the score based on the self-attention mechanism.

$$Q = HW^Q, K = HW^K, W = Softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{2}$$

  where Q and K of dimension d_k denote query and key and W^Q and W^K are parameter matrices for query and key respectively.

2.4 Spectral Layer

Spectral convolution is widely used in time series prediction tasks because of its remarkable ability to learn potential representations of multiple time series in the spectral domain. We first apply a Graph Fourier Transform (GFT)[7] to transfer the graph $\mathcal{G}$ into a representation of the spectral matrix. The fundamental operator is the normalized graph Laplacian, which can be computed as $L = I_N - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, where $W \in \mathbb{R}^{N\times N}$ is the adjacent matrix and D is the diagonal degree matrix with entry $D_{ii} = \sum_j W_{ij}$. And then the Laplacian is diagonalized by the graph Fourier basis $U = [u_0, u_1, \ldots, u_{N-1}] \in \mathbb{R}^{N\times N}$ such that $L = U\Lambda U^T$ where $\Lambda \in \mathbb{R}^{N\times N}$ is the diagonal matrix of eigenvalues.

For the multivariate time-series $\mathbf{X} \in \mathbb{R}^{N\times T}$, GFT is defined as $\mathcal{GF}(X) = U^T X$, and the inverse Graph Fourier Transform (iGFT) is $\mathcal{GF}^{-1}(X) = UX$. This transformation enables the following convolutions, which will be discussed later. And finally, it will be transferred back to the temporal domain by an iGFT for the prediction. But before the spectral convolution, we apply an extra **Intra-series Layer** to further capture the intra-series patterns.

2.5 Intra-series Layer

In this layer, we feed each time series of the Fourier space achieved by GFT into a Discrete Fourier Transform (DFT). The DFT transfers the data from the temporal domain to the frequency domain, while the iDFT does the inverse. Each series is transformed into a series of values arranged in a sequence of frequencies, representing the components of each frequency. Then we apply a 1D Convolution and Gated Linear Units (GLU)[8] to learn the patterns in the periodic data and the auto-correlation features within each series in the frequency domain. The operations can be formulated as:

$$\mathrm{GLU}\left(\theta\left(\hat{X}\right),\theta\left(\hat{X}\right)\right)=\theta\left(\hat{X}\right)\odot\sigma\left(\theta\left(\hat{X}\right)\right) \tag{3}$$

where θ is the convolution kernel, $\odot$ is the Hadamard product, $\hat{X}$is from the output of DFT, and the nonlinear sigmoid gate σ determines the information flow. Afterwards, an inverse Discrete Fourier Transform (iDFT) transfers the data back to the space of Fourier basis achieved by GFT so that we can apply the **Spectral Graph Convolution**.

2.6 Spectral Graph Convolution

The essence of the spectral graph convolution is to filter the spectral representation by a graph convolution operator with learnable kernels. We introduce the spectral graph convolution[4]:

$$y_j=\textstyle\sum_i \Theta_{i,j}(L)X=\sum_i \Theta_{i,j}(\Lambda_i)U^T X_i \tag{4}$$

where the $\Theta_{i,j}$ is the graph convolution kernel corresponding to the ith input and the jth output channel. And recall that from the Spectral Layer, L is the graph Laplacian, Λ is the eigenvalue matrix, and U^T is the GFT operator. So far, our **Spectral Layer** has captured all the effective inter-series relationships and intra-series features. Now the new representation is transformed back to the temporal domain for the prediction.

2.7 Temporal Layer

We apply a fully-connected layer (FC) to perform a mapping from K-dimension to H-dimension, where K denotes the window size of historical data used for forecast and H denotes the window size of future to be predicted. To improve the representation capability, we utilize the GLU again as a non-linearity[8], to selectively pass information through the network.

3 Experiments

3.1 Case Study 1: Stock Price Prediction

Motivation The financial market's volatility is inherently complex and nonlinear, making it challenging to rely solely on a trader's expertise and instinct for analysis and decision-making. Therefore, an intelligent, scientific, and efficient method for guiding stock trading is essential. In this study, we present a hybrid GNN-GLU model for predicting stock prices in China's A-share market.

Data Sources and Preprocessing We obtained daily individual stock data for the China A-share market from CSMAR and company filing data covering the period from December 31, 2019, to March 31, 2023. We classified the stocks into six industry categories: Finance, Properties, Industrials, Conglomerates, Commerce, and Utilities. We selected from the stocks with top 30% market capitalization to ensure model stability.

We chose the closing price on a given trading day as the input for the model. In our experiment, we selected nine stocks with historical data of 500 trading days. We trained the GNN-GLU model based on these data and used a rolling strategy with a window size of 30 to forecast the next 10 days' prices. More specifically, after we forecast the price of day t, we combined the forecasting value with the prices of the past 29 days to predict the price of day $t+1$.

Results and Discussion Our experiment results yielded three main findings. First, our hybrid GNN-GLU model outperforms other methods in terms of accuracy indicators, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Second, our model can roughly capture the trend of stock price (shown in Figures 4, 5, 6 and 7), thus exhibiting potential profitability in financial applications. Through numerous experiments, we constructed various trading strategies that generate positive returns, the details of which are provided in the appendix. Third, our model demonstrates high interpretability connected with reality. The illustration of the matrix W in Figure 3 reflects the strength of the relationship between different stocks, which can help people better understand the dynamics of the stock market.

Ablation Study To better understand the effectiveness of the Graph Convolution Network and correlation matrix W on forecasting, we conducted an ablation study. The results in Table 2 show the effectiveness of correlation generations and the advantage of convolution on the spectral domain to capture inter-series correlation

3.2 Case Study 2: COVID-19 Forecast

Data Collection and Preprocessing We conducted another experiment on newly confirmed COVID-19 cases to verify our model based on real-world data. We chose 10 countries for 200 days(from January 30, 2020 to August 16, 2020).

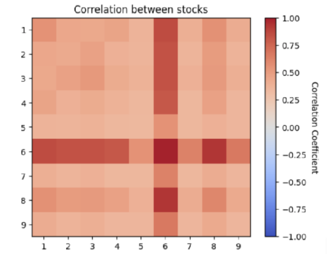

Figure 3: Correlation Matrix

	MAE	RMSE	MAPE(%)
GNN-GLU	**12.32**	**18.35**	**11.30**
LSTNet	14.32	20.35	15.31
FC-LSTM	18.02	27.12	18.74
SVM	22.36	35.11	28.25
RF	21.33	29.26	25.73

Table 1: Performance of different methods in stock price forecasting

	GNN-GLU	w/o GConv	w/o W
MAPE %	**11.3**	14.2	15.3
RMSE %	**18.35**	23.32	28.12
MAE %	**12.32**	18.12	20.70

Table 2: Results for ablation study of the stock dataset

We collected data from COVID-19 Database provided by John Hopkins University. In our experiment, we use the first 140 days for training and the rest 60 days for testing.

Graph Formulation based on Geometric Relationship In this experiment, we proposed a novel approach to constructing a correlation matrix W. By intuition, compared to distant countries, the epidemic spread in two bordering countries was more common. Motivated by this observation, we constructed W in the following way: if two countries are adjacent, their corresponding weight in the matrix is assigned a value of 1 (e.g. Singapore and Thailand), otherwise, it is assigned a value of 0 (e.g. Singapore and Germany). Employing this method, we constructed the correlation matrix in Figure 4.

Result and analysis Through the GNN-GLU model with the predefined matrix W, we forecasted the new cases of COVID-19 in each country. Performance(MAPE%) of different models on COVID-19 is shown in Table 2. We found that our model outperformed others. Moreover, our model also shows an advantage of interpretability since we constructed the correlation matrix based on geographic location.

$$
\begin{array}{r}
\text{Singapore} \\ \text{Thailand} \\ \text{Germany} \\ \text{France} \\ \text{Brazil} \\ \text{Peru} \\ \text{Russia} \\ \text{Italy} \\ \text{Mexico} \\ \text{USA}
\end{array}
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
$$

Figure 4: Correlation Matrix

Methods	MAPE(%)
GNN-GLU	**21.8**
TCN	22.75
LSTNet	23.31
FC-LSTM	25.74
SVM	51.25
RF	57.73

Table 3: Performance of Different Methods on COVID-19 Dataset

4 Conclusion

In this project, we propose a new forecasting model GNN-GLU to take advantage of inter-series correlations and temporal dependencies by shuttling between the temporal domain and spectral domain and capturing corresponding features. We also proposed three approaches to generate the correlation matrix, which is proven to have a significant impact on performance. The results show that GNN-GLU outperforms other forecasting models in the database of the stock market and COVID-19. Moreover, GNN-GLU exhibits strong interpretability regarding the generation and capture of correlation among time series. There are three main directions for further work. First, We could try more methods in the correlation feature extraction process. Second, we may employ residual connections to stack more blocks in our model to improve forecasting accuracy. Third, we will look for its application to more real-world scenarios and try some large-scale datasets.

References

[1] Z. Hamad and I. Abdulrahman, "Deep learning-based load forecasting considering data reshaping using matlab\simulink," *International Journal of Energy and Environmental Engineering*, vol. 13, no. 2, pp. 853–869, 2022.

[2] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[3] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[4] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[5] T. Chebbi and A. Derbali, "The dynamic correlation between energy commodities and islamic stock market: analysis and forecasting," *International Journal of Trade and Global Markets*, vol. 8, no. 2, pp. 112–126, 2015.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[8] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 766–17 778, 2020.

A Appendix: Trading Strategies and Examples

This appendix provides detailed examples of various trading strategies based on the predicted stock prices obtained from our hybrid GNN-GLU model. The strategies were tested over ten days and involved buy and sell options.

A.1 Trading Strategies

A.1.1 Buy Option

If, at some point in time, there is a price less than (or close to) the minimum value of our forecast, then buy; if not, no transaction will be made.

A.1.2 Sell Option

If, at some point in time, there is a price greater than (or close to) the maximum value of our forecast, sell; if not, take the closing price of the last day as the final price to calculate the yield.

A.2 Experimental Examples

Below we provide examples of different situations encountered in the experiment. From Figure 5 to Figure 8:

A.2.1 Example 1

The low predicted in this example is about 182.5, and the high predicted is about 197. We find that on 2020-11-02, the price is below the low, so we buy at 182.5. Since the true price does not exceed the high, we sell at 185 on the last day, yielding a return of approximately 1.37%.

A.2.2 Example 2

The low predicted in this example is about 210, and the high predicted is about 295. We find no actual price below 210, so we do not trade. (Here, we also observe that our model successfully predicted the same overall downward trend as reality, making the decision not to trade a wise one.)

A.2.3 Example 3

The low predicted in this example is about 225, and the high predicted is about 245. We found that on 2021-03-31, the price was below the low, so we chose to buy at 225. We also find that on 2021-04-02, the price is above the high, trading at about 250, yielding a return of approximately 11.11%.

A.2.4 Example 4

The low predicted in this example is about 90, and the high predicted is about 97. We found that on 2020-06-05, the price was below the low, so we chose to buy at 90.5. Since the true price does not exceed the high, we decide to sell at 89.5 on the last day, yielding a return of approximately -1.10%.

After many experiments, we get a positive return of +2.46%.

B Correlation Analysis

From Figure 9: Here we find that the W matrix reflects different strong and weak relationships in different regions. For example, the relationship between Guizhou Maotai and Luzhou Laojiao or Jinshiyuan is stronger, while the correlation between Guizhou Maotai and Yanghe share is weaker.

There exists fundamental proof. For example, Luzhou Laojiao and Guizhou Maotai are high-valued brands with similar market positioning, while Yanghe shares have a relatively lower status in investors' minds. This makes the two receive different degrees of attention in the stock market. Therefore, our correlation study results can find a lot of basis in reality, further confirming our model's interpretability.

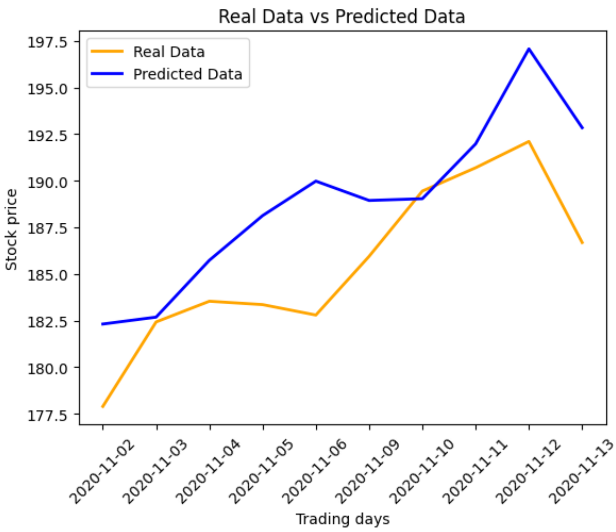

Figure 5: Example 1

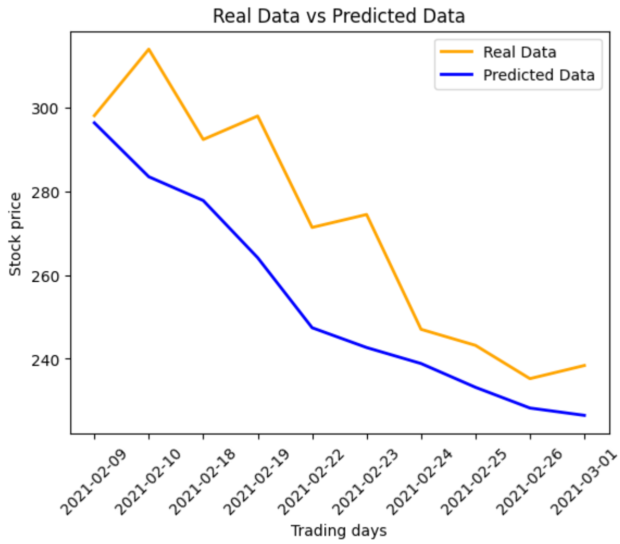

Figure 6: Example 2

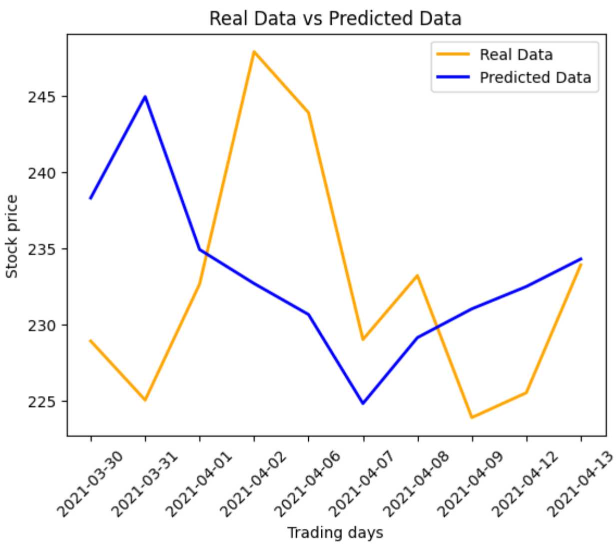

Figure 7: Example 3

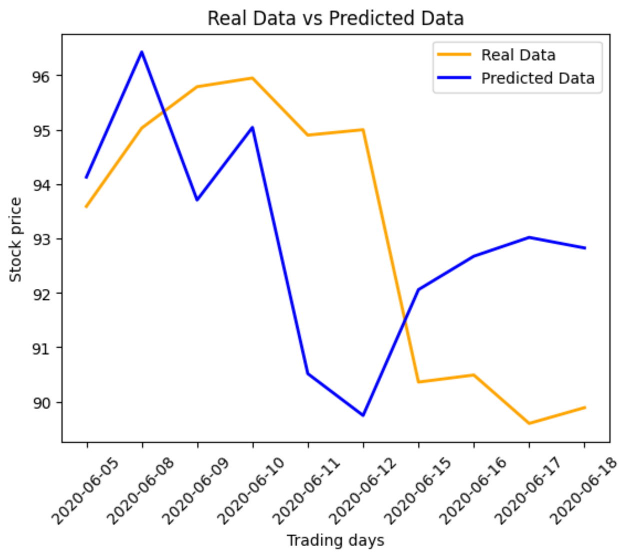

Figure 8: Example 4

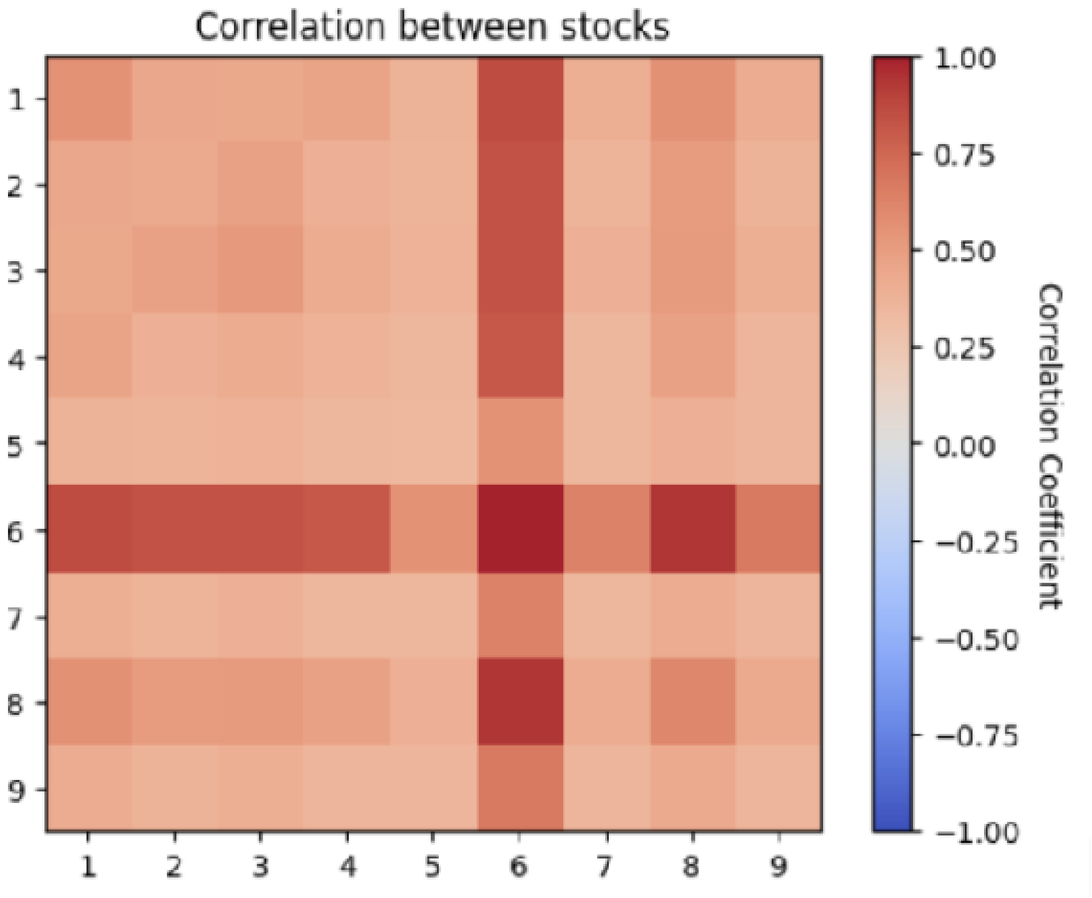

Figure 9: correlation matrix