


## Article

# Stock Market Analysis Using Time Series Relational Models for Stock Price Prediction

Cheng Zhao <sup>1</sup> , Ping Hu <sup>2</sup>, Xiaohui Liu <sup>2</sup>, Xuefeng Lan <sup>3</sup> and Haiming Zhang <sup>4,\*</sup><sup>1</sup> School of Economics, Zhejiang University of Technology, Hangzhou 310023, China<sup>2</sup> College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China<sup>3</sup> Informatization Office, Zhejiang University of Technology, Hangzhou 310023, China<sup>4</sup> Students' Affairs Division, Guangdong University of Petrochemical Technology, Maoming 525000, China

\* Correspondence: zhk2923029@gdupt.edu.cn

**Abstract:** The ability to predict stock prices is essential for informing investment decisions in the stock market. However, the complexity of various factors influencing stock prices has been widely studied. Traditional methods, which rely on time-series information for a single stock, are incomplete as they lack a holistic perspective. The linkage effect in the stock market, where stock prices are influenced by those of associated stocks, necessitates the use of more comprehensive data. Currently, stock relationship information is mainly obtained through industry classification data from third-party platforms, but these data are often approximate and subject to time lag. To address this, this paper proposes a time series relational model (TSRM) that integrates time and relationship information. The TSRM utilizes transaction data of stocks to automatically obtain stock classification through a K-means model and derives stock relationships. The time series information, extracted using long short-term memory (LSTM), and relationship information, extracted with a graph convolutional network (GCN), are integrated to predict stock prices. The TSRM was tested in the Chinese Shanghai and Shenzhen stock markets, with results showing an improvement in cumulative returns by 44% and 41%, respectively, compared to the baseline, and a reduction in maximum drawdown by 4.9% and 6.6%, respectively.

**Keywords:** stock price prediction; stock relationship; time series; long short-term memory; graph convolution neural networks

**MSC:** 68T07

**Citation:** Zhao, C.; Hu, P.; Liu, X.; Lan, X.; Zhang, H. Stock Market Analysis Using Time Series Relational Models for Stock Price Prediction. *Mathematics* **2023**, *11*, 1130. <https://doi.org/10.3390/math11051130>

Academic Editor: Yuanbo Qiao

Received: 26 January 2023

Revised: 20 February 2023

Accepted: 21 February 2023

Published: 24 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

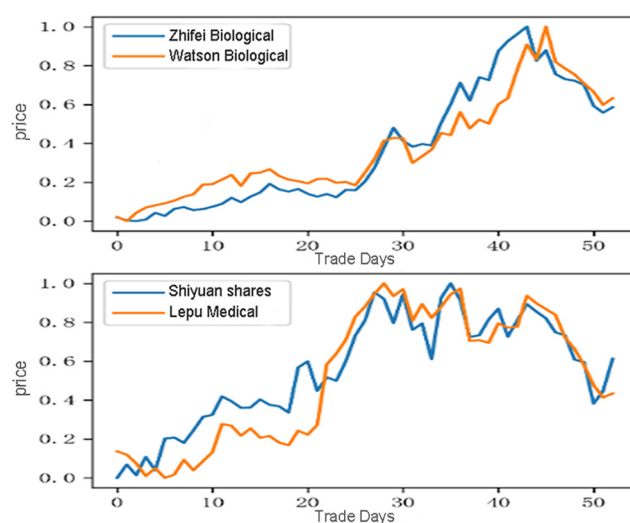
## 1. Introduction

The price of a stock is affected by various factors, such as the macro economy, industry development, enterprise operation, and investments, and it fluctuates at a high frequency. Stock price data have a low signal-to-noise ratio, which always makes predicting stock prices a challenging task. Much research on this topic is based on the characteristic changes in stock prices [1–5]. These prices have two typical characteristics: a time series [5–7] and change co-movement [8,9]. Stock prices change over time in a typical time series, and the theoretical basis for technical analysis of stock market forecasting is the assumption that history repeats itself. At the same time, there is a co-movement between the price fluctuations of multiple related stocks, which may be manifested in simultaneous price rises and falls (for example, two stocks in the same industry face good or bad news), or reverse ups and downs (such as when good news for one stock is bad news for another, competitive stock). The time series of stocks has been widely studied [8,10–14]. However, there is a lack of research on the impact of stock correlation on stock price forecasting.

Many researchers use the time series of stock prices to divide the historical transaction data of a stock into a fixed time interval sequence and input them into a recurrent neural network or various variant models that process the data sequence to identify the changing

pattern of the stock price and thus predict the trend [15–17]. However, this approach treats stocks as isolated individuals, ignoring the fact that they exist in a large and interconnected market, making the trained model less capable of analyzing the entire market and more susceptible to short-term sentiment.

In practice, fund managers usually use their financial knowledge to analyze the stock industry or the concept of the stock, using the correlation between stocks to solve this problem. Although computers have long been used in stock analysis, research on how to use the relationships between stocks to enhance the analytical power of computers is still in its infancy. Figure 1 (data from JoinQuant [18]) shows the influence of the stock relationship on stock price by normalizing the closing prices of two stocks in the same period. The industry relationship is the most typical stock relationship. For instance, Zhifei Biological and Watson Biological belong to the pharmaceutical vaccine industry, and the changes in their stock price are highly synchronized. Industry relations are simple and intuitive and easy to obtain from third-party data platforms but there are other relationships between stocks, such as competitive relationships, and upstream and downstream relationships in the supply chain. Moreover, there may be some short-term implicit relationships between stocks. For example, in Figure 1, Shiyuan shares are located among home office enterprises, while Lepu Medical is located among pharmaceutical vaccine enterprises. Demand for them increased at the same time because of COVID-19, and their fluctuation in price during the epidemic shows a strong correlation. These stock relationships are difficult to obtain fully from third-party data platforms.



**Figure 1.** Stock price correlation.

To address the problem, an approach utilizing a K-means model can be employed to perform automated stock classification and to derive the corresponding stock relationships. These relationships can then be integrated with the time series information obtained from long short-term memory (LSTM) and fed into a graph convolutional network (GCN) to extract relevant information for predicting stock prices. Therefore, we combined time series information and rich relational information to predict stock prices. The main contributions of this paper to the literature are as follows:

- (1) The correlation does not rely on third-party data platforms and uses only stock transaction data to generate stock relationships, as it is based on the K-means model;
- (2) We designed an innovative time series relational model (TSRM) model based on the self-generated stock relationship to integrate both the time series and relationship information to forecast the stock price;
- (3) We verified the universal applicability of the TSRM model via simulated investment experiments in China's A shares in the Shanghai and Shenzhen stock market.

The remainder of the paper is organized as follows: Section 2 summarizes the work and leads to our thoughts; Section 3 describes the structure of the model; and Section 4 describes the experiment-related studies. The results and analysis are discussed in Section 5, and Section 6 presents the conclusions and future work of the paper.

## 2. Related Work

Current research shows that the variant long short-term memory (LSTM) of a recurrent neural network can effectively extract the time series information of stocks and performs well in predicting stock prices. Among these studies, Fischer et al. [12] pioneered the use of LSTM models to predict stock prices based on their time series nature. They used the S&P 500 index of constituent stocks as the stock pool. Using a backtesting experiment, they found that LSTM was significantly more accurate than a comparison model that did not use a sequence memory function in terms of yield and Sharpe ratio (SR). Examples of the latter are the random forest algorithm, deep neural networks and logical classification, methods. Kim et al. [19] combined LSTM with several generalized autoregressive conditional heteroskedasticity models to predict the volatility of stock price indices and experimentally showed that their model was able to obtain lower mean square error (MSE) than others. Teng et al. [13] found that local information over a short period has a significant impact on stock price fluctuations, so they proposed a multi-scale local cue and hierarchical attention-based LSTM model (MLCA-LSTM) to capture the potential trend and price patterns using four different local descriptors to mitigate the noise fluctuations of stock prices over a short period. Aiming at the characteristics of high nonlinearity and the instability of stock data, Cao et al. [10] first used the empirical mode decomposition (EMD) method to decompose stock sequence data into intrinsic modes of different time scales, and then input them into the LSTM model. Experiments show that this LSTM enhancement model combined with EMD is more effective than a simple LSTM, support vector machine, or multi-layer perceptron in predicting financial data. Chen's [11] forecasting experiments in the Hong Kong stock market show that the use of attention mechanisms can enhance the LSTM model's prediction of stock prices.

Integrating stock relationships into stock price forecasting is a relatively new research direction. In the US NASDAQ and NYSE markets, Feng et al. [20] obtained stock industry classification information from the Wikidata platform, arguing that stocks in the same industry classification have the same industry relationship. All stock pairs in the same industry form a topological graph representing stock relationships and are input into the GCN to extract stock relationships and predict stock prices. The experimental results show that the stock price forecast using GCN exceeds the traditional time series model in terms of return. Chen et al. [21] obtained stock relationships from the Wind platform and used the relationship transfer method to increase the number of stock relationships. Experiments show that increasing the number of stock relationships can increase the performance of predicting stock prices.

Based on the above literature, in this paper, we chose LSTM to extract the stock time series information and GCN to extract the stock relationship information. However, there are some shortcomings in using the industry classification provided by third-party platforms to obtain stock relationships, namely:

- (1) Third-party platform data updates are not timely;
- (2) Relationship data are not comprehensive and the researcher can obtain only the same industry relationship;
- (3) No platform supporting all markets exists.

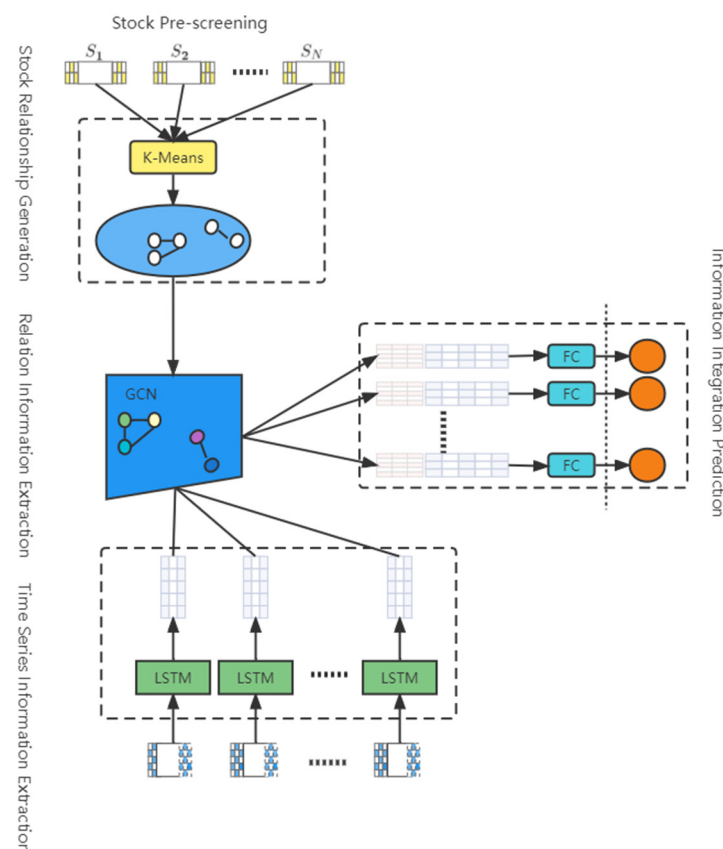
In response to these issues, based on stock trading data, we used the K-means model to automatically generate stock classification and derive stock relationships. This avoids the use of industry classification provided by third-party platforms and provides a timely, consistent, and more adequate method for obtaining stock correlation.

### 3. Stock Temporal Relation Model

The symbols used in this article are as follows:  $x$ ,  $X$ , and  $\mathcal{X}$  represent vectors, matrices, and tensors, respectively, and  $x_t^i = (x_1, x_2, \dots, x_n)^T$ ,  $X_i = (x_1^i, x_2^i, \dots, x_t^i, \dots, x_{T_s}^i)$ ,  $\mathcal{X} = (X_1, X_2, \dots, X_N)$ .  $X_t^I$  represents the trading characteristics of stock  $i$  on trading day  $t$ ,  $X_i$  represents the time series of stock  $i$  on trading day  $T_s$ , which is used to extract time series information,  $n$  is the stock feature dimension,  $T_s$  represents the length of time series, and  $N$  represents the number of stocks. In addition,  $S_i = (c_1^I, c_2^I, \dots, c_t^I, \dots, c_{T_R}^I)^T$  is used for stock clustering as the stock price feature of the stock, where  $c_t^i$  represents the closing price of stock  $i$  on trading day  $t$ .  $C_1$ ,  $C_2$ ,  $C_3$  represent the hidden layer dimensions in LSTM and GCN, respectively.

#### 3.1. Model Structure

The proposed TSRM model structure is shown in Figure 2. The original trading data of the stock, after certain filtering conditions, output the three-dimensional stock data  $\mathcal{X}$  for the experiment.



**Figure 2.** Time series relational model.

According to Figure 2, during the experimental period, the time series of trading characteristics ( $X_1, X_2, \dots, X_i, \dots, X_N$ ) of all stocks on  $T_s$  day were input into LSTM to obtain the time series characteristics of each stock. Similarly, the closing price series ( $S_1, S_2, \dots, S_i, \dots, S_N$ ) of all stocks on the  $T_R$  day before the experimental period was input into the K-means stock relationship generation module to cluster and generate stock relationships. Then, the topological graph was generated by the stock relationship, and the time series characteristics of the stock were used as an expression of the nodes in the graph, which were input into GCN to obtain the influence of the relevant stock on a stock. Finally, the time series characteristics and relationship impact factors of the stock were

spliced together and input into the fully connected layer to obtain the stock price predicted for the next day, and the stock with the largest predicted return was selected.

### 3.2. Stock Pre-Selected

This paper sets up two filtering conditions to filter stocks in the market: [20]

- (1) There is no suspension day;
- (2) The stock price is not less than 15 RMB.

The first screening condition avoids excessive damage to the statistical characteristics of the data, thus preventing the model from learning false information. The second screening condition was used to avoid excessive volatility of small stocks that change rates at extreme values, also affecting the overall learning model. The stock pre-selection stage returned  $N$  stocks for the experiment.

### 3.3. Extraction of Timing Information

The output of the hidden layer of the last sequence element of the LSTM [22] was utilized as the timing feature of the stock. LSTM is an excellent variant of recurrent neural network, which uses four layers of neural network in one recurrent unit to solve gradient disappearance and gradient explosion problems during the training of long sequences.

$$h_t^i = LSTM(x_t^i, h_{t-1}) \quad (1)$$

Of which,  $h_t^i \in R^{C_1}$ . The sequence characteristics of all stocks are combined into  $H \in R^{N \times C_1}$ .

### 3.4. Stock Relation Generation

The actual relationships that exist among stocks, such as industry and upstream and downstream relationships, are referred to as entity relationships. A stock may have entity relationships with many other stocks and these relationships interact and influence each other. As shown by Formula (2), the combined relationships ultimately act on stock trading behavior, especially stock prices, and cause the prices of these stocks to exhibit certain correlations. Thus, the similarity between stock prices reflects a combination of many relationships between stocks. There is a kind of mutual mapping between the entity relationships between stocks and the correlation between stock prices,

$$\begin{aligned} (R_{i,j}^1, R_{i,j}^2, \dots, R_{i,j}^N) &\xrightarrow{f} r_{i,j}^k \\ (R_{i,j}^1, R_{i,j}^2, \dots, R_{i,j}^N) &\xleftarrow{g} r_{i,j}^k \end{aligned} \quad (2)$$

where  $R_{i,j}^n$  denotes that stock  $i$  and stock  $j$  have some entity relationship, and  $r_{i,j}$  denotes the correlation between stock  $i$  and stock  $j$  share prices, both of which are denoted by 0 and 1 in Boolean data.

The stock price correlation can be obtained by stock price clustering. The relationship between stocks based on stock price clustering is called a clustering relationship in this paper. The K-means clustering relationships [23] of stocks were employed to replace their entity relationships and to model the impact of stock relationships on stock returns. K-means is a popular clustering algorithm used in unsupervised machine learning. The goal of K-means is to group a set of data points into a predefined number ( $K$ ) of clusters based on their similarity to each other. The stock relationship is a unidirectional topological graph, where the nodes represent stocks and the lines between the nodes represent the relationship between two stocks. In the following, the trading data of stocks is processed step by step to obtain the two-dimensional adjacency matrix,  $A_{kmeans}$ , representing the topology of stock clustering relationships.

- (1) The closing price series  $S_i = (c_1, c_2, \dots, c_{T_R})$  of a stock on a trading cycle are collected as its stock price characteristics to calculate the similarity between stock prices, and

the maximum–minimum normalization method is used to eliminate the effect of price differences across stocks:

$$c_t = \frac{c_t - \text{Min}(c_1, c_2, \dots, c_{T_R})}{\text{Max}(c_1, c_2, \dots, c_{T_R}) - \text{Min}(c_1, c_2, \dots, c_{T_R})} \quad (3)$$

- (2) The appropriate number of categories is selected to determine the  $K$  value.
- (3) The initial cluster center is determined. As shown by Formula (4), the Euclidean distance formula is utilized to quantify distances due to its simplicity and effectiveness in handling numerical stock price data. Its capability to discern similar patterns, such as comparable price fluctuations, enables efficient clustering [24]. The first stock  $S_0$  in the stock list is taken as the first initial clustering center point  $P_0$ , and then the point farthest from that point is selected as the second initial clustering center point, and then the point with the largest nearest distance from the first two points is selected as the center point of the third initial cluster, and so on, until all  $K$  initial clustering centers are selected.

$$\text{dis}(S_i, P_j) = \|S_i, P_k\|^2 = \sqrt{\sum_{t=1}^T (c_t^{S_i} - c_t^{P_k})^2} \quad (4)$$

- (4) The stock series ( $S_1, S_2, \dots, S_I$ ) and the initial clustering centers ( $P_1, \dots, P_K$ ) are input into the K-means model for clustering, as can be seen in Formulas (5) and (6), iterating to minimize the objective function:

$$J = \sum_{i=1}^N \sum_{k=1}^K r_{ik} \|S_i - P_k\|^2 \quad (5)$$

$$r_{ik} = \begin{cases} 1, & S_i \in k \\ 0, & S_i \notin k \end{cases} \quad (6)$$

where  $r_{ik}$  denotes 1 when stock  $S_i$  is otherwise classified into category  $k$  and 0.  $P_k$  denotes the mean vector of category  $k$ .

- (5) Based on the division of stocks into categories, the set of stock relationships  $R$  is inferred and the adjacency matrix  $A_{kmeans}$  used to represent the topology of stock relationships is generated, as shown in Formula (7).

$$R = \{(s_i, s_j) | s_i, s_j \in k\} \quad (7)$$

The main problem of clustering relationships is that different choices of  $K$  value will introduce an invalid relationship. For example, when  $K$  takes the extreme value of 1, all stocks are included in the classification and there is a clustering relationship between every and any pair of stocks, leading to numerous meaningless relationships. When the  $K$  value increases gradually, the number of stock classifications increases and the number of stocks in the same classification decreases, resulting in a decrease in the number of stock clustering relationships. However, the increase in  $K$  value requires a limit to avoid too many classifications separating the stocks that, in fact, belong to one industry.

### 3.5. Relation Information Extraction

The GCN requires two inputs: an adjacency matrix  $A$  that represents the relationship and a matrix  $H$  that represents the temporal characteristics of the stock. A simplified form of GCN proposed by Kipf et al. [25] was utilized to obtain the impact of stock relations. As can be seen in Formulas (8) and (9), this consists of two layers of graph convolution for the input layer to the hidden layer and the hidden layer to the output, respectively.

$$\hat{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}} \quad (8)$$



$$Y = f(H, A) = \text{softmax}(\hat{A} \text{ReLU}(\hat{A}HW^{(0)})W^{(1)}) \quad (9)$$

where  $\hat{A} = D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}}$  is the adjacency matrix representing the undirected graph  $G$  which adds self-notice, and  $D \in R^{(N, N)}$  is the degree matrix of  $G$ .  $W^{(0)} \in R^{C_1 \times C_2}$  is the weight matrix from the input layer to the hidden layer,  $W^{(1)} \in R^{C_2 \times C_3}$  is the weight matrix from the hidden layer to the output layer, and  $H \in R^{N \times C_1}$  is the temporal feature of the output of the LSTM model.  $Y \in R^{N \times C_3}$  is used as the output of the relation module to indicate that the stock in focus is influenced by the stock in question.

### 3.6. Information Fusion Forecasting

The LSTM output, which expresses stock time series information, and the GCN output (see Formula (10)), which expresses the impact of stock relationships, are integrated using a weight matrix to predict future returns,

$$z = \text{ReLU}(w[h, y]^T + b) \quad (10)$$

where  $h$  is the hidden layer expression of stock at the last node of the LSTM model,  $y$  is the stock's output of the GCN model influenced by the underlying stock, and  $z$  is the expected return of the stock on the next day, represented here by  $\frac{o^t}{o^{t+1}}$ , with  $o^t$  being the opening price on day  $t$ . This is because the simulated trading strategy used in this paper also involves buying and selling at the opening price.

The Adam algorithm was used to minimize the loss of the training data. The parameters of the LSTM model and the GCN model are optimized in minimizing the loss function.

## 4. Experimental Setup

### 4.1. Experimental Data and Experimental Environment

The constituent stocks of the Shanghai and Shenzhen 300 indices were used as the stock pool because these stocks are representative of market capitalization and account for more than 60% of the total market capitalization of the domestic stock market. Based on the screening criteria, 87 stocks were screened in Shanghai and 68 stocks were screened in Shenzhen [18].

The experimental dates selected in this paper range from 1 January 2019 to 1 September 2020 and the experimental time is divided into three intervals: the training set (1 January 2019 to 1 January 2020), the validation set (1 January 2020 to 1 March 2020), and the test set (1 March 2020 to 1 September 2020). Six types of intraday trading data were collected as stock characteristics: [26] the opening price, the closing price, high price, low price, volume, and turnover rate, because they are typical, common, and easily accessible, and any stock exchange will provide these six types of data. The observed time series data is processed through depolarization, missing value filling, normalization, and neutralization, followed by the construction of features and labels using a sliding window approach [27].

The proposed method was implemented with Python 3.7 and PyTorch 1.12.1, using a GeForce RTX 3080Ti graphics card and CUDA version 11.6. Parallel computation was performed on the GPU via the DataParallel function of PyTorch 1.12.1. The system configuration included an Intel Core i9-10850K processor operating at 3.60 GHz, 64 GB of RAM, and Windows 10 as the operating system. The software tool used was PyCharm, Professional Edition 2020.3.2, provided by JetBrains.

### 4.2. Model Parameters and Trading Strategies

When extracting the time series information, the number of LSTM layers was determined to be 2, the dimensions were set to 64 and 128, and the time step  $T$  of the time series was set to 20, i.e., one trading month, through several experiments.

When extracting the relationship information, the time period  $T_R$  of the closing price series of the obtained stock relationship was set to one trading year, and the dimensions of the input layer to the hidden layer and the hidden layer to the output layer of GCN were

set to 128 and 128, respectively. During training, the number of iterations of the model was set to 1000, the learning rate was set to 0.0001, and the dropout mechanism was added to prevent overfitting. The dropout parameter was set to 0.2.

The model backtesting experiment selected the stocks with the highest predicted returns from the next day's predictions for all stocks and uses a daily cycle-based "buy-hold-sell" trading strategy to simulate the market investment:

When the stock market opens on trading day  $t$ : the trader uses the model to get the return ranking of all stocks in the stock pool and then buys the stock with the highest return ranking.

When the stock market opens on trading day  $t + 1$ : the trader sells all the stocks bought on day  $t$ .

#### 4.3. Evaluation Indicators

The purpose of this study was to accurately predict the return on stocks while balancing the return and risk of asset investments; therefore, the following five evaluation metrics were used to evaluate and compare the models: [28,29] MSE, mean absolute error (MAE), the cumulative investment–return ratio (IRR), maximum drawdown (MDD), and SR.

The equations and related descriptions are in Table 1. Among them, MSE and MAE are common evaluation metrics for regression tasks in machine learning, and they are often used to measure the difference between the predicted and true values of a model. Therefore, the smaller the MSE and MAE, the smaller the difference between the predicted and true values of the model, and the higher the prediction accuracy. IRR is the main evaluation metric that sums the stock's returns on each trading day to obtain the final return for the test cycle. MDD describes the worst-case scenario that can occur during the investment process and significantly influences investors' pessimism. It is thus an important risk metric. Finally, SR evaluates the performance of an investment behavior from both return and risk dimensions in a comprehensive manner and is one of the most frequently used evaluation metrics in stock trading. A good model of stock price prediction will result in smaller MSE, MAE, and MDD, as well as larger IRR and SR [27,30].

**Table 1.** Related evaluation indicators.

Metrics	Formula	Description
MSE	$\frac{1}{N} \sum_{t=1}^N (g_t - p_t)^2$	$g_t$ : true value $p_t$ : predicted value
MAE	$\frac{1}{N} \sum_{t=1}^N  g_t - p_t $	$g_t$ : true value $p_t$ : predicted value
IRR	$\sum_{t=1}^N \frac{O_t - O_{t-1}}{p}$	$o_t$ : open price of day $t$ $p$ : the principal
MDD	$\max\left(\frac{r_{up} - r_{down}}{r_{up}}\right)$	$r_{up}/r_{down}$ : a local high/low point of return ratio
SR	$\frac{r_p - r_f}{\delta_p}$	$r_p$ : return of portfolio $r_f$ : risk-free rate $\delta_p$ : standard deviation of the portfolio's excess return

## 5. Experimental Results and Comparative Analysis

### 5.1. Comparison Model

This study applied the K-means algorithm to analyze daily stock trading data and combined relationship information from GCN with time series data from LSTM. This integration enables the prediction of future stock prices and supports investment decision-making for investors. Therefore, the experiments were designed to answer the following questions:

To what extent can the stock relationships extracted by the K-means algorithm using stock trading data cover the benchmark stock relationships extracted from third-party data platforms? How are the K values for the K-means algorithm determined?

We selected the China Securities Regulatory Commission (CSRC) stock industry classification to obtain the benchmark relationships because the CSRC is the official authority for analyzing the Chinese stock market. According to the attribution of the experimental stocks in the CSRC stock industry classification, 230 relationships were extracted from the exper-



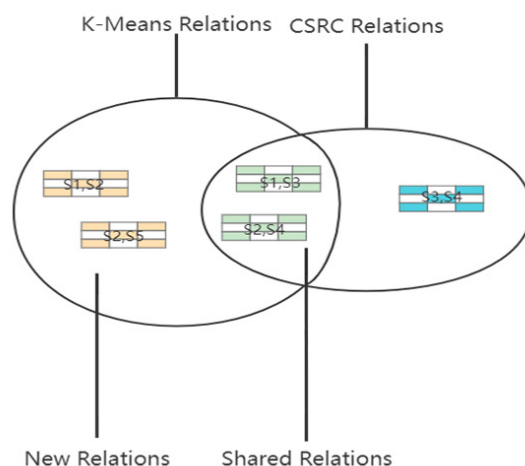
imental stocks in Shanghai and 191 relationships were extracted from the experimental stocks in Shenzhen.

Does the TSRM model make full use of relational data to improve prediction performance and can stock market backtest returns of the TSRM model outperform the benchmark LSTM model?

We selected the LSTM model, which has been widely used by previous studies, the TSRM-K-means model, which is based on the K-means to obtain stock relationships, and the TSRM-CSRC model, which is based on the CSRC data platform to obtain stock relationships, for backtesting experiments. Meanwhile, the SSE index and SZSI index were used as the backtest benchmarks for the Shanghai and Shenzhen markets, respectively.

### 5.2. The Effect of the K Value on the Coverage of Clustering Relationship

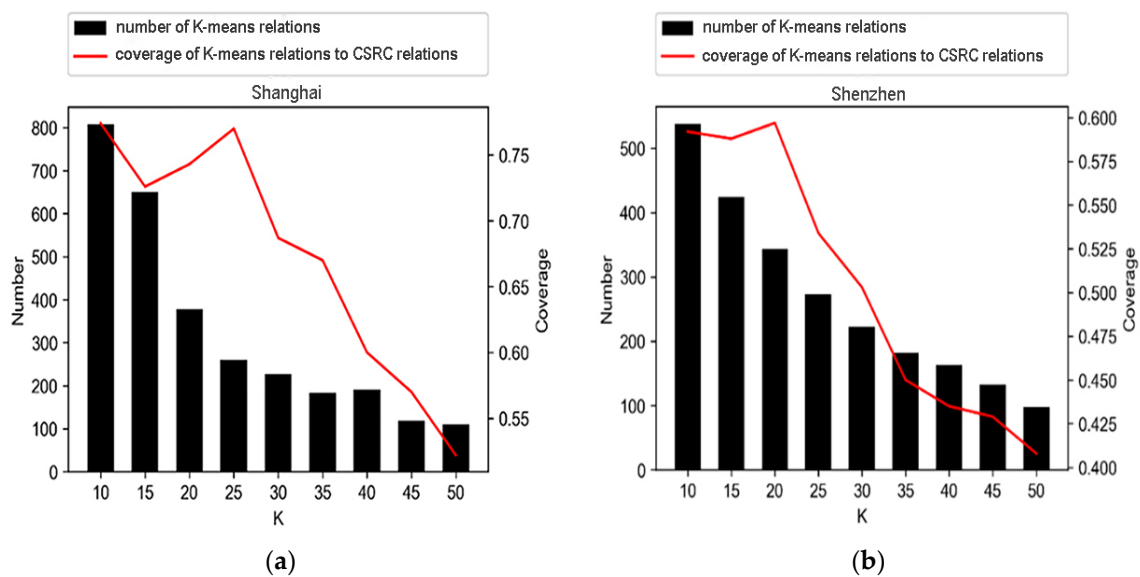
To facilitate the observation of relationship coverage, this paper defines a set of four stock relations: CSRC relations, K-means relations, shared relations, and new relations. Among them, the stock relationship extracted according to the stock industry classification of the CSRC is called the CSRC relationship; the stock classification is obtained by the clustering algorithm, and the stock relationship extracted from it is called the clustering relationship. All stock relationships that belong to both the CSRC relationship and the clustering relationship are called common relationships; all the stock relationships that only belong to the clustering relationship and do not belong to the CSRC relationship are called new relationships. Figure 3 shows the range of these four sets of relationships.



**Figure 3.** Four kinds of sets for stock relations.

The K-value selection criterion is used to reduce the number of clustering relations generated while covering, as much as possible, the benchmark relations provided by some authoritative third-party platforms.

As can be seen from Figure 4 and Table 2, the total number of clustered relationships decreases gradually as the value of  $K$  increases, but the coverage of clustered relationships to CSRC relationships, i.e., the number of shared connections, among them, maintains a certain degree of stability at the beginning. After a turning point (in Shanghai when  $K = 25$ , and in Shenzhen when  $K = 20$ ), the number of shared relationships decreases rapidly. This may be due to too large of a  $K$  value, which leads to too many classifications of stock and destroys the classification pattern of stocks, separating stocks that should be in the same category. Therefore, the turning point where the number of common relationships is significantly reduced is selected as the optimal  $K$  value of the K-means algorithm for classifying stocks automatically and extracting the relationship topology for the subsequent GCN model.



**Figure 4.** (a) The number of K-means relations and its coverage ratio to third-party relations (CSRC) in Shanghai markets under different K values; (b) The number of K-means relations and its coverage ratio to third-party relations (CSRC) in Shenzhen markets under different K values.

**Table 2.** Changes in clustering relationships in Shanghai and Shenzhen markets using different K values.

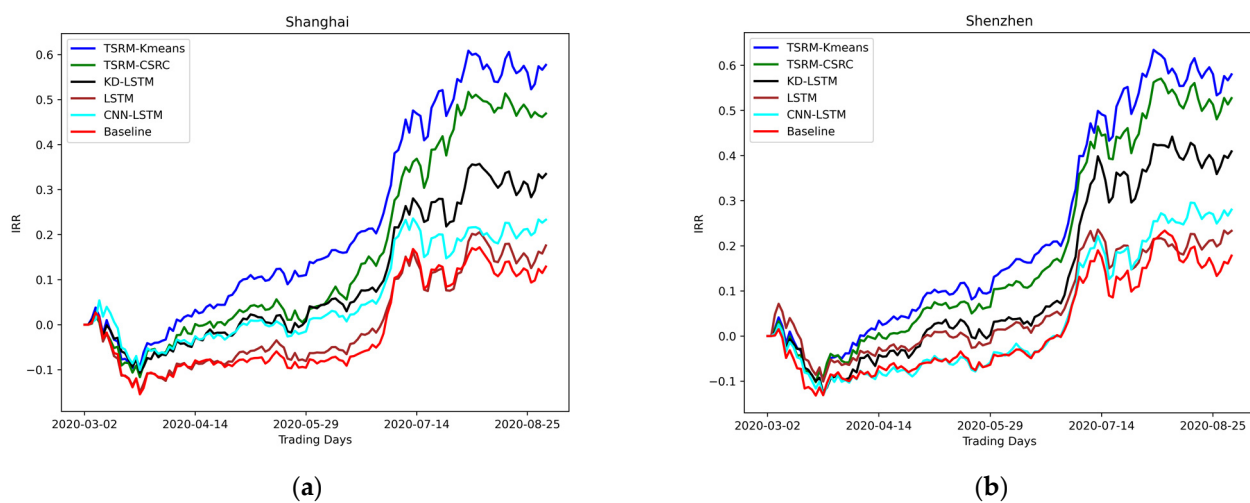
Market	K	Cluster	Common	New
Shanghai	10	986	178	808
	15	817	167	650
	20	548	171	377
	25	436	177	259
	30	386	158	228
	35	337	154	183
	40	329	138	191
Shenzhen	10	538	425	113
	15	424	327	112
	20	344	230	114
	25	273	171	102
	30	222	126	96
	35	182	96	86
	40	163	80	83

### 5.3. TSRM Backtesting Return Study

To evaluate the TSRM-K-means model with optimal K determination and a comparison model, backtesting experiments were conducted.

As can be seen in Figure 5, the TSRM-K-means obtains the largest cumulative returns in both markets and is larger than the cumulative returns obtained by the comparison model for most of the tested time periods.

As can also be seen from Table 3, TSRM-K-means has the smallest MSE and MAE, the largest SR, and the smallest maximum retracement, indicating that the TSRM-K-means model, which integrates relational and time series information, can significantly improve prediction accuracy, reduce return volatility, and obtain a good return-to-risk ratio.



**Figure 5.** (a) Investment–return ratio (IRR) of different models in Shanghai markets; (b) Investment–return ratio (IRR) of different models in Shenzhen markets.

**Table 3.** Comparison of models' backtesting experimental.

Market	Model	MSE	MAE	IRR	MDD (%)	SR
Shanghai	TSRM-K-means	0.000513	1.632	0.57	6.5	3.72
	TSRM-CSRC	0.000586	1.663	0.46	7.6	3.38
	KD-LSTM [31]	0.000631	1.717	0.33	8.1	2.89
	LSTM [22]	0.000779	1.838	0.18	10.3	2.45
	CNN-LSTM [32]	0.000725	1.814	0.23	9.6	2.52
	Baseline	-	-	0.13	11.4	1.53
Shenzhen	TSRM-K-means	0.000487	1.614	0.58	6.8	3.70
	TSRM-CSRC	0.000562	1.672	0.52	7.8	3.26
	KD-LSTM [31]	0.000675	1.745	0.41	8.3	2.97
	LSTM [22]	0.000788	1.851	0.23	12.2	2.13
	CNN-LSTM [32]	0.000703	1.820	0.28	10.5	2.39
	Baseline	-	-	0.17	13.4	1.66

## 6. Limitations

Although the TSRM model is a great improvement over the benchmark model in financial market analysis and forecasting, there are some limitations:

- (1) The large volume of data and variable market styles; the correlation can only be analyzed based on recent data.
- (2) This study did not incorporate external data, such as industry or concept data or macroeconomic indicators, to supplement the analysis. Incorporating such data may provide additional insights into the dynamics of the stock market.

## 7. Conclusions and Future Work

This study has proposed a novel model TSRM that integrates temporal information of stock prices and relationships among stocks to improve stock price prediction. By leveraging LSTM to capture temporal information and GCN to extract relationships among stocks, the TSRM model has shown significant improvement in cumulative returns and maximum drawdown when compared to traditional LSTM models in the Chinese Shanghai and Shenzhen stock markets.

Experimental results have indicated that the K-means algorithm is indeed capable of extracting correlations between stocks from daily trading data, and these relationships can facilitate improved stock price predictions. This contribution fills a gap in the current

literature and provides a more comprehensive perspective on stock prices and relationships, which is an important guide to decision-making and trading strategies in financial markets.

Future work includes expanding the model by incorporating more dimensions of data in the stock clustering process to further enhance the understanding of relationships among stocks. This could involve exploring alternative clustering algorithms or the inclusion of additional data sources, such as news sentiment analysis or macroeconomic indicators.

In summary, the TSRM model proposed in this study has demonstrated its effectiveness for improving stock price prediction by leveraging both temporal and relational information. We are confident that the contributions of this study will lead to new avenues for research and practical applications in the field of stock market analysis.

**Author Contributions:** Formulating the idea, C.Z., P.H. and X.L. (Xiaohui Liu); methodology, C.Z., P.H. and X.L. (Xuefeng Lan); theory, C.Z. and P.H.; algorithm design, C.Z. and P.H.; result analysis, X.L. (Xiaohui Liu); writing, P.H.; reviewing the research, C.Z., X.L. (Xuefeng Lan) and H.Z.; supervision; X.L. (Xuefeng Lan); project administration, C.Z.; funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Major Humanities and Social Sciences Research Projects in Zhejiang higher education institutions, grant number 2023QN082 and the National Natural Science Foundation of China, grant number 61902349.

**Data Availability Statement:** The Shenzhen Stock and Shanghai Stock Datasets presented in this study are available at JoinQuant <https://www.joinquant.com/data> (accessed on 26 February 2022), reference number [18].

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

TSRM	time series relational model
EMD	empirical mode decomposition
MLCA	multi-scale local cue and hierarchical attention-based LSTM model
CSRC	China Securities Regulatory Commission
LSTM	long short-term memory
GCN	graph convolutional network
MSE	mean square error
MAE	mean absolute error
IRR	cumulative investment–return ratio
SR	Sharpe ratio
MDD	maximum drawdown

## References

1. Dai, W.; Shao, Y.E.; Lu, C.-J. Incorporating feature selection method into support vector regression for stock index forecasting. *Neural Comput. Appl.* **2012**, *23*, 1551–1561. [CrossRef]
2. Fama, E.F.; French, K.R. International Tests of a Five-Factor Asset Pricing Model. *J. Financ. Econ.* **2015**, *123*, 441–463. [CrossRef]
3. Zhang, J.; Li, L.; Chen, W. Predicting Stock Price Using Two-Stage Machine Learning Techniques. *Comput. Econ.* **2020**, *57*, 1237–1261. [CrossRef]
4. Zhang, J.; Teng, Y.-F.; Chen, W. Support vector regression with modified firefly algorithm for stock price forecasting. *Appl. Intell.* **2018**, *49*, 1658–1674. [CrossRef]
5. Zhao, Y.; Yang, G. Deep Learning-based Integrated Framework for stock price movement prediction. *Appl. Soft Comput.* **2022**, *133*, 10992. [CrossRef]
6. Liu, Z.; Li, Y.; Liu, H. Fuzzy time-series prediction model based on text features and network features. *Neural Comput. Appl.* **2021**, *35*, 3639–3649. [CrossRef]
7. Wang, H.; Zhang, Y.; Liang, J.; Liu, L. DAFA-BiLSTM: Deep Autoregression Feature Augmented Bidirectional LSTM network for time series prediction. *Neural Netw.* **2022**, *157*, 240–256. [CrossRef] [PubMed]
8. Baruník, J.; Kočenda, E.; Vácha, L. Asymmetric Connectedness on the U.S. Stock Market: Bad and Good Volatility Spillovers. *J. Financ. Mark.* **2015**, *27*, 55–78. [CrossRef]
9. Nguyen, V.C.; Nguyen, T.T. Dependence between Chinese stock market and Vietnamese stock market during the COVID-19 pandemic. *Heliyon* **2022**, *8*, e11090. [CrossRef]

10. Cao, J.; Li, Z.; Li, J. Financial time series forecasting model based on CEEMDAN and LSTM. *Phys. Stat. Mech. Its Appl.* **2019**, *519*, 127–139. [CrossRef]
11. Chen, S.; Ge, L. Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. *Quant. Financ.* **2019**, *19*, 1507–1515. [CrossRef]
12. Fischer, T.G.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
13. Teng, X.; Zhang, X.; Luo, Z. Multi-scale local cues and hierarchical attention-based LSTM for stock price trend prediction. *Neurocomputing* **2022**, *505*, 92–100. [CrossRef]
14. Zhao, C.; Liu, X.; Zhou, J.; Cen, Y.; Yao, X. GCN-based stock relations analysis for stock market prediction. *PeerJ Comput. Sci.* **2022**, *8*, e1057. [CrossRef]
15. Md, A.Q.; Kapoor, S.; AV, C.J.; Sivaraman, A.K.; Tee, K.F.; Sabireen, H.; Janakiraman, N. Novel optimization approach for stock price forecasting using multi-layered sequential LSTM. *Appl. Soft Comput.* **2022**, *134*, 109830. [CrossRef]
16. He, H.; Dai, S. A prediction model for stock market based on the integration of independent component analysis and Multi-LSTM. *Electron. Res. Arch.* **2022**, *30*, 3855–3871. [CrossRef]
17. Widiputra, H.; Mailangkay, A.; Gautama, E. Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. *Complexity* **2021**, *2021*, 9903518. [CrossRef]
18. JoinQuant. Available online: <https://www.joinquant.com/data> (accessed on 26 February 2022).
19. Kim, H.Y.; Won, C.H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Syst. Appl.* **2018**, *103*, 25–37. [CrossRef]
20. Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; Chua, T.-S. Temporal Relational Ranking for Stock Prediction. *ACM Trans. Inf. Syst. (TOIS)* **2019**, *37*, 1–30. [CrossRef]
21. Chen, Y.; Wei, Z.; Huang, X. Incorporating Corporation Relationship via Graph Convolutional Neural Networks for Stock Price Prediction. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018.
22. Nelson, D.M.Q.; Pereira, A.M.; Oliveira, R.A.d. Stock market's price movement prediction with LSTM neural networks. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1419–1426.
23. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [CrossRef]
24. Wu, D.; Wang, X.; Wu, S. Construction of stock portfolios based on k-means clustering of continuous trend features. *Knowl. Based Syst.* **2022**, *252*, 109358. [CrossRef]
25. Kipf, T.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907.
26. Alfonso, G.; Ramirez, D.R. A Nonlinear Technical Indicator Selection Approach for Stock Markets. Application to the Chinese Stock Market. *Mathematics* **2020**, *8*, 1301. [CrossRef]
27. Wang, C.; Chen, Y.; Zhang, S.; Zhang, Q. Stock market index prediction using deep Transformer model. *Expert Syst. Appl.* **2022**, *208*, 118128. [CrossRef]
28. Chen, W.; Zhang, H.; Mehlawat, M.K.; Jia, L. Mean-variance portfolio optimization using machine learning-based stock price prediction. *Appl. Soft Comput.* **2021**, *100*, 106943. [CrossRef]
29. Wang, W.; Li, W.; Zhang, N.; Liu, K. Portfolio formation with preselection using deep learning from long-term financial data. *Expert Syst. Appl.* **2020**, *143*, 113042. [CrossRef]
30. Wang, H.; Wang, T.; Li, S.; Zheng, J.; Guan, S.; Chen, W. Adaptive Long-Short Pattern Transformer for Stock Investment Selection. In Proceedings of the International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022.
31. Chen, Y.; Wu, J.; Wu, Z. China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach. *Expert Syst. Appl.* **2022**, *202*, 117370. [CrossRef]
32. Aldhyani, T.H.H.; Alzahrani, A.A.M. Framework for Predicting and Modeling Stock Market Prices Based on Deep Learning Algorithms. *Electronics* **2022**, *11*, 3149. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.