Published: 10 October 2024
Online AM: 19 September 2024
Accepted: 12 September 2024
Revised: 11 April 2024
Received: 25 January 2023

SURVEY

A Systematic Review on Graph Neural Network-based Methods for Stock Market Forecasting

MANALI PATEL, S. V. National Institute of Technology, Surat, GJ, India

KRUPA N JARIWALA, S. V. National Institute of Technology, Surat, GJ, India

CHIRANJOY CHATTOPADHYAY, FLAME University, Pune, MH, India

A Systematic Review on Graph Neural Network-based Methods for Stock Market Forecasting

MANALI PATEL, Sardar Vallabhbhai National Institute of Technology, Surat, India
KRUPA JARIWALA, Sardar Vallabhbhai National Institute of Technology, Surat, India
CHIRANJOY CHATTOPADHYAY, FLAME University, Pune, India

Financial technology (FinTech) is a field that uses artificial intelligence to automate financial services. One area of FinTech is stock analysis, which aims to predict future stock prices to develop investment strategies that maximize profits. Traditional methods of stock market prediction, such as time series analysis and machine learning, struggle to handle the non-linear, chaotic, and sudden changes in stock data and may not consider the interdependence between stocks. Recently, graph neural networks (GNNs) have been used in stock market forecasting to improve prediction accuracy by incorporating the interconnectedness of the market. GNNs can process non-Euclidean data in the form of a knowledge graph. However, financial knowledge graphs can have dynamic and complex interactions, which can be challenging for graph modeling technologies. This work presents a systematic review of graph-based approaches for stock market forecasting. This review covers different types of stock analysis tasks (classification, regression, and stock recommendation), a generalized framework for solving these tasks, and a review of various features, datasets, graph models, and evaluation metrics used in the stock market. The results of various studies are analyzed, and future directions for research are highlighted.

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Semantic networks**; • **Networks** → Network economics; • **Mathematics of computing** → **Graph algorithms;**

Additional Key Words and Phrases: Graph neural networks, stock market forecasting, non-euclidean data, relational modeling, temporal and spatial dimension

ACM Reference Format:
Manali Patel, Krupa Jariwala, and Chiranjoy Chattopadhyay. 2024. A Systematic Review on Graph Neural Network-based Methods for Stock Market Forecasting. *ACM Comput. Surv.* 57, 2, Article 34 (October 2024), 38 pages. https://doi.org/10.1145/3696411

1 Introduction

After the COVID-19 pandemic, a large number of the population has been concerned about making investments to put their money to work and efficiently build wealth. Equity or stock investment has recently gained more popularity compared to other financial instruments, such as fixed deposits,

Manali Patel and Krupa Jariwala contributed equally to this research.

Authors' Contact Information: Manali Patel, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India; e-mail: manali20101995@gmail.com; Krupa Jariwala, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India; e-mail: knj@coed.svnit.ac.in; Chiranjoy Chattopadhyay, FLAME University, Pune, Maharashtra, India; email: chiranjoy.chattopadhyay@flame.edu.in.

ACM 0360-0300/2024/10-ART34
https://doi.org/10.1145/3696411

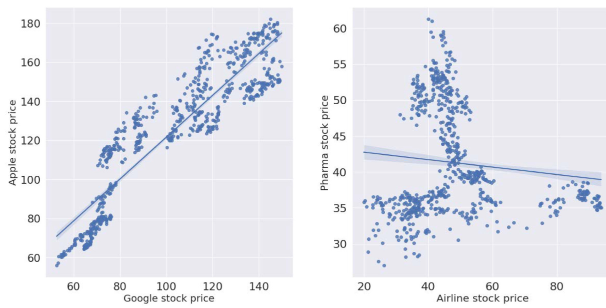
mutual funds, and government bonds, as it gives another source of income in terms of dividends and capital gains with a minimum amount of initial investment. With an accurate forecast of the equity market, investors can upscale their profits. At the same time, the equity market exhibits highly non-linear, volatile, and chaotic patterns due to its dependency on a plethora of factors, such as the sentiment and psychology of investors, political affairs, global events, and macroeconomic variables [106]. Modeling such a noisy and complex behaviour of stock data makes prediction tasks more challenging.

A profit earned by any investment instrument is associated with a considerable amount of risk. To reduce risk and increase profitability in the stock market, investors rely on two well-known analytical approaches: Fundamental and Technical analysis approaches [121]. A fundamental analyst aims to find the intrinsic value of a security to decide whether it is undervalued or overvalued by considering company fundamentals such as balance sheets, profit and revenue, and so on, whereas technical analysts study past historical data to build various indicators that predict future trends. According to the **Efficient Market Hypothesis (EMH)** theory, [36] the financial market is unpredictable as it reflects all available information. The direct implication is that no fundamental or technical analysis can provide a competitive advantage. Certain theories [41, 95] disagree with the EMH theory and argue that to some extent the market is predictive, allowing investors to make a reasonable profit with a certain level of risk associated with the investment. This has attracted professionals from academia and finance to build prediction models and develop trading strategies that can yield profits.

1.1 Evolution of Stock Market Forecasting Approaches

This section describes the various approaches adopted for stock market forecasting over a period of time. Predicting the stock market involves forecasting future trends based on patterns found in historical data. This type of problem is known as a time series forecasting task, and it requires a model that can effectively capture the temporal dependencies within the data. There have been many statistical approaches used to try and solve this problem, with the goal of accurately predicting future stock market movements. The success of these approaches often depends on their ability to recognize patterns in the data that may indicate future trends. Statistical or econometric approaches include univariate forecasting models such as **Autoregressive Integrated Moving Average (ARIMA)** [1, 2, 7, 102] and its extended version that can handle the seasonality of time series data known as **Seasonal ARIMA (SARIMA)** [128], as well as the multivariate forecasting model **Vector Auto Regression (VAR)** [11, 13, 141]. To measure the volatility of the stock market, symmetric **Generalized AutoRegressive Conditional Heteroskedasticity (GARCH)** [105, 113] and asymmetric GARCH [85, 97] models are the most widely used statistical approaches for the stock market. All the above statistical models work well for small-scale datasets for short-term prediction but fail to capture the nonlinear dynamic patterns exhibited by stock time series data. Moreover, the assumptions made by models regarding data distribution limit their use in practical scenarios. To overcome the limitations of statistical approaches, traditional machine learning models such as **Support Vector Machines (SVMs)** [17, 83] and Random Forest [62, 108] have been used. A comparison study done by Bhattacharjee and Bhattacharja [14] highlighted the fact that machine learning approaches have an edge over statistical methods. Several studies [76, 115, 124] have provided an extensive review of machine learning approaches for stock prediction tasks. Despite their ability to model nonlinearity, these models are plagued by the curse of dimensions and fail to capture the long-term temporal dependencies of underlying data.

Deep learning (DL), a subset of machine learning that mimics the learning process of the human brain to handle large, complex data, and extract relevant features, has become popular for stock market prediction, leading to improved performance compared to statistical and traditional



(a) Positive correlation. (b) Negative correlation.

Fig. 1. Illustration of Correlation analysis between a pair of organizations.

machine learning models [51, 93]. Advances in computational power and algorithms have made these models more feasible to use. For the stock market, various deep learning models have been proposed, including **Artificial Neural Networks (ANNs)** [46, 119, 148], memory-based **Long Short-term Memory (LSTM)** [38, 101, 112, 139] and its optimized version **Gated Recurrent Unit (GRU)** [44, 130] for the stock market. The ability to process sequential data gives LSTM and GRU models an edge over other models and makes them widely used for forecasting. Apart from this, several researchers have applied a state-of-the-art computer vision model, **Convolutional Neural Networks (CNNs)** [20, 22, 77] to extract the underlying hidden temporal patterns from past data to predict future trends. CNN models designed for prediction tasks are sensitive to the choice of kernel and filter sizes. Jiang [65] and Hu et al. [59] have reviewed deep learning models comprehensively.

Because stock price movement takes into account all available information, publicly available news, company announcements, or investor sentiments, it influences future prices [67, 84, 110, 161]. Based on this theory, another research direction considers **Natural Language Processing (NLP)** approaches that apply deep learning models to perform sentiment analysis on data collected from news articles, Twitter, and online forums to forecast future prices [64, 74]. But the impact of the sentiment data deteriorates over time, and it also takes an uncertain amount of time for investors to react to the news that reflects in the stock price movement [6]. Moreover, a sentiment-based approach is suitable only for short-term prediction and suffers from data scarcity [163] and questionable authenticity of source data.

Previous approaches to stock market prediction have typically treated each stock as an isolated entity, only considering the individual target company. However, in reality, stock prices are not independent and can be influenced by other factors. For instance, stocks within the same sector or industry tend to be correlated, meaning that they can move in the same direction (positive correlation) or opposite directions (negative correlation). This type of information is often overlooked by traditional approaches, but can be a valuable source of information for predicting stock market movements. To prove this point, Figure 1 depicts this positive and negative correlation between stock prices of the same and different sectors, respectively, during the COVID-19 pandemic period of 2019–2022.

As seen in Figure 1(a), Google and Apple, two organizations that belong to the same IT sector, are positively correlated and exhibit similar trends, whereas in Figure 1(b) stocks of the airline and pharmacy sectors are negatively correlated during the COVID-19 pandemic period as there was a sudden decline in the airline industry due to travel restrictions, whereas pharmacy services were in demand. This establishes the fact that there exists some kind of relational dependence (positive or negative) among stocks. By incorporating this relational dependency into the prediction model, the effect of external events (in this COVID-19 situation) is studied more efficiently. If two companies have similar fundamentals or some kind of correlation, and if one company is affected by some external factors, then it will also impact the associated company. This is known as the momentum spillover effect. To study this effect, the stock market should be considered as a network rather than a singular entity. As a result, this introduces relational dimension modeling to the stock prediction task. To explore such spatial or relational dependency, graph neural networks are being applied for stock market prediction, where a network is built by considering various relationships. **Graph Neural Networks (GNN)** models are capable of processing non-Euclidean network data and extracting latent interactions in complex systems. These have been successful in various domains, such as the discovery of an antibiotic [56, 99], recommendation systems [103, 140], traffic prediction [66, 114], and so on. Now, they are also being applied to stock market forecasting. With the incorporation of GNNs, the research question is modelled as **"How efficiently can a model capture the relational as well as temporal dependency hidden in the data?"**

To summarize the above discussion about various approaches, Figure 2 depicts the road map of stock market forecasting approaches being applied over a period of time. The statistical approaches have been replaced by traditional machine learning approaches that have the ability to capture non-linear patterns. A subset of machine learning approaches known as DL techniques have outperformed traditional approaches for stock market prediction due to their self-adaptive and feature extraction abilities. For stock market forecasting, the DL methods are further classified into two categories based on the data representation, i.e., Euclidean and non-Euclidean approaches. The Euclidean temporal DL approaches have been applied to sequential market data or stock images to capture the long range dependency. With the rise of online social platforms such as Twitter, online forums, and financial reports, specialized deep learning approaches have been applied to perform sentiment analysis and study its effect on stock prices. With the advent of advanced DL models to process the non-Euclidean geometry space, knowledge graph analysis-based methods for the stock market have seen a rise in recent years.

Furthermore, to compare stock market forecasting approaches, we have identified the main characteristics of the stock market. These characteristics are: **(1) Non-linearity (NL):** the ability to capture the fluctuations; **(2) Short Temporal Dependency (STD):** the ability to exploit past information within a small time window; **(3) Long Temporal Dependency (LTD):** the ability to capture information that occurs over an extended period; **(4) Feature Engineering (FE):** the requirement of manually defining features; and **(5) Relational adaptability (RA):** the ability to model the complex relationships between stock pairs and extract high-dimensional non-Euclidean features. The approaches and their comparison based on the above discussed characteristics are also presented in Figure 2.

As depicted in Figure 2, knowledge graph analysis offers distinct advantages over alternative approaches. It excels at capturing non-linearity, handling short and long-range dependencies, adapting to relational structures, and eliminating the need for explicit feature engineering. Therefore, processing financial knowledge graphs using specialized deep learning models (Graph Neural Networks) helps to model complex hidden concepts and gives them an edge over other approaches.

1.2 Purpose of This Study and Contributions

A graph neural network-based approach to analyze knowledge graphs is still in its initial stages and needs to be further explored to improve the prediction performance of existing approaches. There have been several surveys regarding graph representation learning and graph modelling techniques in general [34, 142, 143]. But with the recent application of graph neural networks in the stock market, there is a need to review recent progress specific to this domain in detail. To our

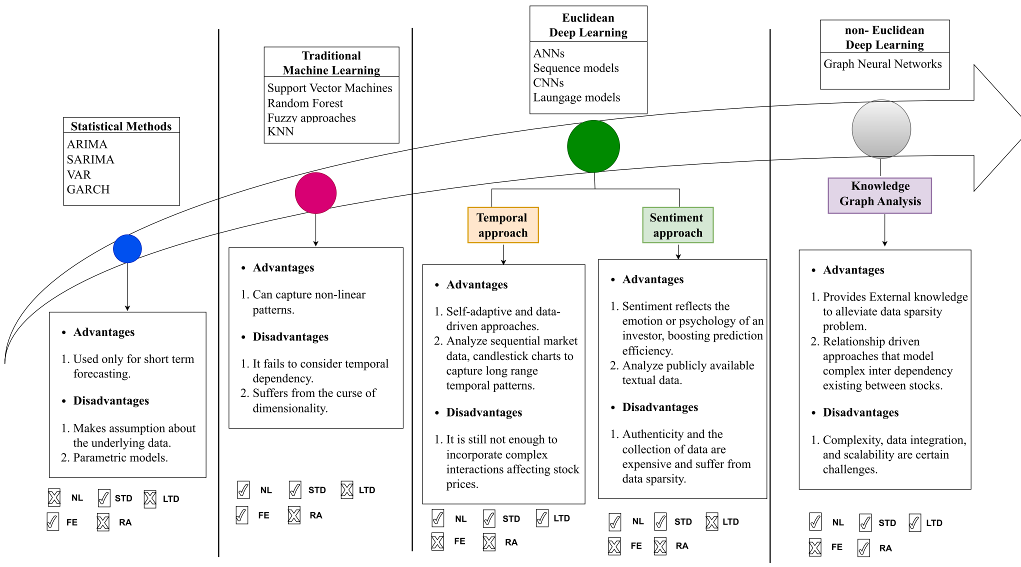


Fig. 2. Evolution of stock market forecasting approaches.

knowledge, prediction models for stock market forecasting based on graph approaches have not been summarized. This work synchronizes the researchers with the recent progress being made in this specific direction. The contribution of this work is summarized as follows:

— This work presents a general framework for stock market forecasting, leveraging graph-based approaches. Furthermore, different modules defined in the framework are discussed in detail.
— This article comprehensively reviews and discusses graph-based prediction models for the stock market, categorized by their specific tasks, such as classification, regression, or stock recommendation.
— This article discusses different data sources, datasets, and evaluation parameters for stock market forecasting.
— The results from each reviewed article are highlighted. A detailed analysis is carried out to perform a comparative assessment of different models.
— It also highlights open issues and future directions to guide further research in this domain.

To avoid any ambiguity, we will use the stock market and financial market interchangeably.

2 Research Efforts

2.1 Search Query

The goal of this research is to examine the use of graph-based modeling techniques for forecasting the stock market, with a particular focus on deep learning on graphs and their application in finance. To identify the most relevant information, we search various scientific research databases, such as the ACM digital library, IEEE Xplore, Springer, Elsevier, MDPI, Taylor & Francis, and Google Scholar. We have also included the pre-print database arXiv to be synchronized with the most recent progress on this topic. A combination of keywords were used in the search, including ["graph neural networks" OR "knowledge graphs" OR "relational modeling"] AND ["stock market prediction" OR "stock market forecasting" OR "stock movement prediction"]. After identifying the various studies published in journals and conferences from the selected databases, we studied their titles and abstracts to exclude the irrelevant ones. Since stock market forecasting is an

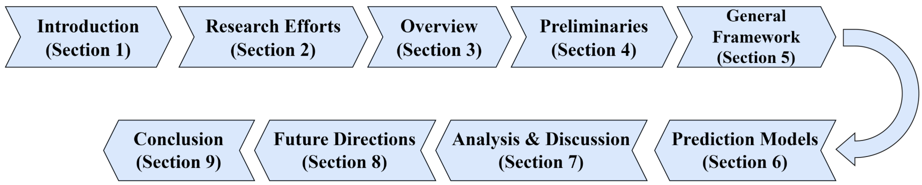


Fig. 3. Summary of sections in this manuscript.

interdisciplinary domain, articles published in the qualitative finance domain that are not related to computer science and engineering are excluded. We followed the below mentioned inclusion and exclusion criteria to choose the studies for final analysis:

Inclusion criteria:

— The study should explicitly discuss or apply graph-based deep learning methods for stock market prediction.
— The study must define the clear objectives of the research, the theoretical background, graph modeling techniques, and quantitative analysis.
— Some key papers defining graph representation and learning techniques to formulate and model stock market graphs are also included.

Exclusion criteria:

— The studies were not written in English.
— Demo papers, short papers, and opinion papers are not included.
— The studies that have not provided a clear presentation or sufficient insights into their approach are excluded.

The roots of the success of graph-based modeling techniques in different real time applications are found in the work [71] done in 2016. Afterwards, different variations of the model are built considering the requirements of underlying applications such as social network analysis, drug discovery, and traffic forecasting. Application of these models to the finance domain has gotten attention over the last five years, but the research is still in its infancy. The review will cover the graph-based modeling techniques applied for stock market prediction from 2016 to 2023.

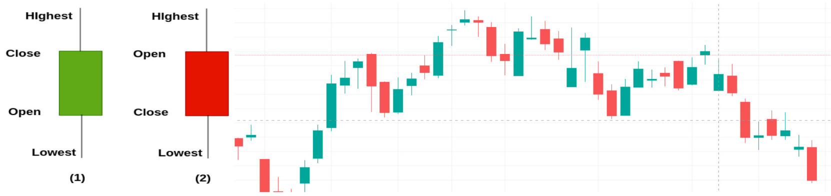
2.2 Organization of Survey Article

This section describes the flow of the survey article. The summary of sections presented in this manuscript is depicted in Figure 3.

Section 3 describes the most commonly used features as well as datasets used for stock market forecasting tasks and their sources. Section 4 clarifies the common notations and definitions used throughout the article. The research question is formulated for different tasks, i.e., classification, regression, and recommendation, that are performed using graph networks. Section 5 discusses a general framework that considers graph-based approaches for prediction. All the relevant modules included in the framework are discussed in detail. Section 6 reviews the graph-based models according to the underlying modeling techniques, features, dataset, and evaluation parameters. Section 7 represents the analysis and discussion on evaluation parameters and various modules presented in Section 5. Section 8 highlights the future directions. Section 9 summarizes the article.

3 Overview

To begin with, this section describes the features and datasets for stock market forecasting in detail. The significance of each feature as well as the characteristics of datasets are explained in Sections 3.1 and 3.2, respectively.



(a) Types of candlesticks. (b) Market view in the form of candlesticks.

Fig. 4. (a) Candlesticks: (1) Bullish candlestick: It is depicted in green; the close price is higher than the open price, indicating an uptrend. (2) Bearish candlestick: It is depicted in red; the close price is lower than the open price, indicating a downtrend. (b) Market view: The image is taken from online platform TradingView (https://in.tradingview.com), which depicts market trends using candlesticks.

3.1 Features for Stock Market Forecasting

The selection of features for the forecasting task is a crucial task due to the dependency of stock data on a plethora of factors. The right combination of features helps capture the volatile nature of financial data and improve prediction accuracy. In this section, we describe the most commonly used features and their sources.

— **Market data or Historical data:** Market data refers to the following elements within the considered time frame, i.e., yearly, monthly, or daily.
(1) **Closing price:** The last traded price of a share.
(2) **Open price:** The price at which the first trade occurred.
(3) **High price:** The highest price of a trade.
(4) **Low price:** The lowest price of a trade.
(5) **Volume:** The number of shares traded.
The historical data is accessible from websites such as Yahoo Finance,[1] Bloomberg,[2] Kaggle,[3] and so on. Among all the elements, the closing price is a prominent market feature as it reflects the market trend. The pictorial representation of market data is known as candlestick patterns. There are two types of candlesticks, namely, Bullish and Bearish representing uptrend and downtrend, respectively. Figure 4(a) represents the Bullish and Bearish candlesticks, and the market view in the form of candlestick patterns is shown in Figure 4(b).

— **Fundamental data:** Fundamental data is information related to a company's financial accounting that is released quarterly throughout the year. Fundamental analysts use this data to find the intrinsic value of a commodity or stock. By analyzing the fundamental data, such as balance sheets, cash flow statements, and income statements, a decision is made as to whether a stock is undervalued or overvalued. If it is undervalued, then there is a possibility that the stock price will go up and a buy recommendation will be generated. Likewise, if it is overvalued, then stock prices will fall, and a sell recommendation will be given. These are used in the process of selecting a stock for long-term investment. The examples include earnings per share, P/E ratio, dividend yield ratio, and return on equity [10, 133].

— **Technical data:** Technical data is derived from historical closing price and volume data and is used to make trading decisions. These are preferred for short-term predictions. Traders use a combination of various technical indicators to build a strategy that generates the buy

[1] https://finance.yahoo.com/
[2] https://www.bloomberg.com/
[3] https://www.kaggle.com/

or sell signal accordingly [29, 35]. The technical indicators are categorized as leading and lagging indicators. Leading indicators are used to anticipate future market directions, and lagging indicators are used to analyze the trend of the market given the past data. These lagging and leading indicators are used to determine entry and exit points in the market by analyzing overbought and oversold conditions. An oversold condition implies that the market has fallen sharply and is expected to rise, whereas an overbought condition implies that the market will fall. Lagging indicators are easier to measure, but they are slow to react, taking longer to see results, so traders cannot book larger gains. Whereas leading indicators react quickly to price changes to anticipate the future but are prone to providing false signals. Technical indicators have an impact on market prediction performance [109]. However, selecting appropriate technical indicators is also critical. Traders use a combination of various technical indicators to anticipate the future price and trend, and it is a subjective choice.

— **Relational data:** This type of data depicts some kind of relationship existing between companies and industries. A knowledge graph is built using this data, which can be directed or undirected. This type of pre-defined relational data is accessible from the WikiData website [132]. It is an open source knowledge graph consisting of over 95 million entities. Specifically, the financial knowledge graph has first order relationships that depict the straightforward relation R between two entities as well as second order relationships that include another entity bridging two entities having different relationships denoted as R_1 and R_2. Basically, first order relation between two stocks A and B is denoted as: $A \xrightarrow{\text{R}} B$ and second order relation is denoted as: $A \xrightarrow{\text{R}_1} B \xleftarrow{\text{R}_2} C$. In WikiData, various instances, classes, and relationships are given unique identifiers. Consider the first order relationship, "Tim Cook is an owner of Apple company," where Tim Cook (Q312) and Apple company (Q265852) are instances, and they are associated with the relation "owner of" (P1830), where the unique identifier is shown in ( ). This association is extracted using SPARQL query service provided by the WikiData services. Haller et al. [49] have provided an explanation of entities and classes included in the database as well as the query process in detail. Utilizing this service, various relationships are extracted and a financial knowledge graph is built.

— **Textual data:** Textual data refers to news articles, PTT forums, news headlines, tweets, social media, blogs, and so on. This data is mainly used to study the effect of sentiment on prediction tasks. For stock prediction, this data is crawled from sources such as Reuters,[4] Twitter,[5] and Bloomberg.[6] Textual data, along with market data, is used for stock market forecasting. One of the challenges faced is the analysis of unstructured, vast amounts of text data. Fataliyev et al. [37] review textual data-based methods used for stock market analysis. Modeling textual data requires proper pre-processing and advanced models that can capture the semantics of the underlying textual data [4].

Market, fundamental, and technical data are stock specific features that define statistical trends and fluctuations in stock prices. The textual data is analyzed using sentiment approaches to predict the future. While the relational data that represents the interconnections between stock pairs is used to build a financial knowledge graph. The graph approaches process this and incorporate market, fundamental, technical, or textual data as node features.

[4] https://www.reuters.com/
[5] https://twitter.com/
[6] https://bloomberg.com

Table 1. Notations Used to Describe the Preliminaries

Symbol	Description	Symbol	Description
N	Number of vertices/companies	V	Set of vertices
D	Dimension of input features	E	Set of edges
S	Stock symbol	$g(.)$, $h(.)$	Activation functions
p	Look back window size	$f(.)$	User defined spatio-temporal function
c	Number of classes	$R = \{r_1, r_2, \ldots, r_k\}$	Set of relationships
k	Number of relationships	$x_t^S \in R^D$	Feature vector of stock S at time t.
T	Future horizon	$X_{(t)} \in R^{N\times D}$	Feature matrix of N stocks at time t
t	Time instance	$X_t^S \in R^{p\times D}$	Historical sequence for stock S at time t considering look back window of size p
$return_t^S$	Return ratio of stock S at time t		
$close_t^S$	Closing price of stock S at time t	$F_t \in R^{N\times p\times D}$	Feature tensor

3.2 Dataset

The datasets may refer to the stocks of a single company or to a market index, which is a collection of stocks chosen based on certain criteria, such as market capitalization or industry. The market capitalization of a company is calculated by multiplying the number of outstanding shares by the current stock price, and companies are often classified as "blue chip," "large cap," "mid cap," or "small cap" based on their market capitalization values. The stock exchange of a country typically creates a market index that reflects the overall economic situation of the country. The market capitalization value is an indicator of volatility; a higher value indicates a stable market, while a lower value indicates a volatile market [75]. The details about the various market indexes and market capitalization values of various countries are provided in the supplementary material.

4 Preliminaries

Before delving into the details of the graph neural networks for the stock market, first we describe some notation and definitions to reduce the ambiguity throughout the article. Table 1 summarizes the symbols used throughout the article.

Definition 4.1 (Graph). $G = (V, E)$ denotes a graph, where V is a set of N vertices, each of which is a company, and E is the set of edges connecting two vertices v_i and v_j that have some relationship such that $e_{ij} =< v_i, v_j >$.

Definition 4.2 (Node Specific Spatio-temporal Graph). A node specific spatio-temporal graph is represented as $G_{(t)} = (V, E, X_{(t)})$ where V and E are defined in the graph definition above. $X_{(t)} \in R^{N\times D}$ represents a dynamically changing feature matrix of N nodes. D represents the number of features for a node.

Definition 4.3 (Adjacency Matrix). An adjacency matrix denoted as $A \in R^{N\times N\times k}$ encodes the relationship between vertices; in the case of a single relationship, k is set to 1. A stock graph can be static or dynamic, undirected or directed, weighted or unweighted, single or multi-relational. Based on this, an adjacency matrix A is built. In Section 5.1, graph construction techniques are discussed in detail.

Definition 4.4 (Historical Sequence). A historical sequence of a stock S at time instance t contains D-dimensional input features of previous p data points and is denoted by $X_t^S = \{x_{t-p+1}^S, \ldots, x_t^S\} \in R^{p\times D}$, where $x_t^S \in R^D$

Definition 4.5 (Feature Tensor). The feature tensor for N stocks at time instance t is denoted as $F_t = \{X_t^1, X_t^2, \ldots, X_t^N\} \in R^{N\times p\times D}$, representing the number of features of N stocks from the last p days.

4.1 Problem Formulation

In this section, we formulate a stock market forecasting problem with respect to graph neural networks. Our goal is to learn a user-defined function $f(.)$ that can capture both spatial and temporal dependency. We have classified various tasks that are performed utilizing graph data structures for the financial market. These tasks are described below:

(1) Classification task: The objective of a classification task is to predict the movement of a stock, i.e., up (1), down (−1), or neutral (0). The movement is calculated from two consecutive closing prices, $close_{t-1}$ and $close_t$ as

$$m_t = \begin{cases} 1 & \text{if } close_{t-1} < close_t \ , \\ -1 & \text{if } close_{t-1} > close_t, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The classification task predicts future stock movement, denoted as matrix $\hat{M}_{t+1} \in R^{N \times c}$, given a feature matrix F_t, a graph matrix $G_{(t)}$, and an adjacency matrix A at time instance t, where c is the number of classes. Equation (2) is used to represent the classification task:

$$\hat{M}_{t+1} = f(F_t, G_{(t)}, A). \tag{2}$$

(2) Regression task: The goal of the regression task is to predict the future closing price or return value of N stocks for a horizon of T length given a feature tensor F_t, a graph $G_{(t)}$, and an adjacency matrix A at time instance t. The return ratio for stock S, denoted by $return_t^S$, is calculated as

$$return_t^S = \frac{close_t^S - close_{t-1}^S}{close_{t-1}^S}. \tag{3}$$

A regression task's result is represented as a matrix $\hat{Y} \in R^{N \times T}$, where T is equal to 1 in the case of single-step ahead prediction:

$$\hat{Y}_{t+T} = f(F_t, G_{(t)}, A). \tag{4}$$

(3) Stock recommendation task: As a starting point, we define a return ratio for stock S, denoted by $return_t^S$ (refer to Equation (3)). The goal of the stock recommendation task is to predict the ranking score of each stock denoted as $ret\hat{u}rn_{t+1} = \{ret\hat{u}rn_{t+1}^1, ret\hat{u}rn_{t+1}^2, \ldots, ret\hat{u}rn_{t+1}^N\} \in R^N$ given a feature matrix F_t, $G_{(t)}$, and an adjacency matrix A at time instance t. From this list, the most profitable stocks are recommended for purchase. Equation (5) is used to express this mathematically:

$$ret\hat{u}rn_{t+1} = f(F_t, G_{(t)}, A). \tag{5}$$

All the above-mentioned tasks are defined for edge-specific static graphs, where the adjacency matrix A does not change over time. The adjacency matrix of edge-specific dynamic graphs is also a function of time, denoted as $A(t)$.

5 General Framework

In this section, we describe a general framework for graph-based stock market forecasting approaches. The task is to capture the spatial as well as the temporal dependency hidden in the non-Euclidean data. From the reviewed literature, we have first identified the common pattern adopted by researchers for prediction using a graph-based approach. Based on that, we have segregated this task and defined three different modules; these are: **(1) Graph Construction Module**, to construct a graph from relational data. **(2) Historical Information Encoder**, to capture temporal dimension. Various deep learning models are used as historical information encoder, which are discussed in the subsequent section. **(3) Relational Module**, to capture spatial dependence. GNNs

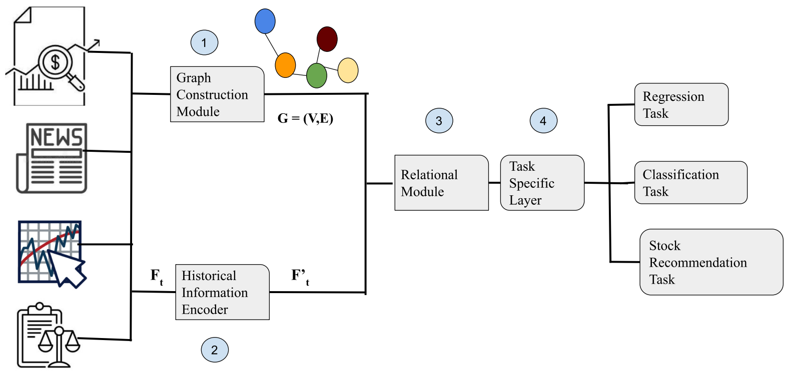


Fig. 5. General Framework for graph-based stock market forecasting approaches. ① A graph is constructed by considering market, relational, or textual data. ② The temporal dimension is exploited using this module, and an updated feature tensor is obtained. ③ To exploit the spatial dimension, the relational module is given an adjacency matrix and updated features. ④ This module projects the output of the relational module into a task-specific format.

and their variants are used to model relational dimensions. Each module is discussed in detail in a subsequent section. Figure 5 depicts the general framework for stock market prediction utilizing graph neural networks. As seen in Figure 5, it is common practice to use historical information encoders first to exploit the temporal dependency, followed by a relational encoder module that uses graph neural networks to learn the spatial dimension. The order of the historical information encoder and relational module is interchangeable.

5.1 Graph Construction Module

In this section, we describe various methods to construct a financial market graph. It is constructed using one of the methods mentioned below:

5.1.1 Statistical Graph Construction Method. This method considers the statistical descriptor to measure the association between two entities. The most commonly used measure is Pearson's correlation coefficient, denoted as $corr$. It ranges from [−1,1] with −1 representing negative correlation, 1 denoting positive correlation, and 0 showing no correlation. An edge is considered between the stock pairs having a positive or negative correlation, where edge weight is a correlation value. Consider a stock pair $< S_i, S_j >$ and its closing price series at time instance t denoted as $S_i(t)$ and $S_j(t)$, respectively, where $S_i(t) = \{close_i(t), close_i(t-1), \ldots, close_i(t-p+1)\}$ and $S_j(t) = \{close_j(t), close_j(t-1), \ldots, close_j(t-p+1)\}$, $close_i(t)$ represents the closing price of stock S_i at time instance t, and p is previous periods to be considered for correlation calculation. Equations (6) and (7), respectively, are used to calculate the mean and variance of a closing price series:

$$\hat{S}_i(t) = \frac{1}{p}\Sigma_{k=0}^{p-1}[close_i(t-k)], \tag{6}$$

$$Var[S_i(t)] = \frac{1}{p}\Sigma_{k=0}^{p-1}[close_i(t-k) - S_i\hat{(t)}]^2. \tag{7}$$

Using Equation (8), we now define Pearson's correlation coefficient:

$$corr < S_i, S_j >_t = \frac{\Sigma_{k=t-p+1}^{t}[close_i(k) - S_i\hat{(t)}][close_j(k) - S_j\hat{(t)}]}{\sqrt{(var[S_i(t)])(var[S_j(t)]) \times p}}. \tag{8}$$

Adjacency matrices are created by considering the correlation value between each stock pair calculated from the above formula. To avoid over-smoothing, the adjacency matrix A is binarized by specifying a threshold value [81, 153]. An edge is considered between two stocks if the correlation between them $corr < S_i, S_j >$ is greater than the user-defined threshold value. Thus, the graph generated is unweighted. To build a weighted graph, the absolute value of the correlation coefficient is considered.

The strong correlation between two variables can be the result of the influence of the third variable. To overcome this problem, a partial correlation coefficient is calculated that considers the influence of the third variable [69]. If two stocks S_i and S_j are affected by another stock S_k, then the partial coefficient is calculated using Equation (9). For simplicity, we have not included the notations t and p in the equation:

$$partial - corr < S_i, S_j, S_k >= \frac{corr < S_i, S_j > -corr < S_i, S_k > corr < S_j, S_k >}{\sqrt{(1 - corr < S_i, S_k >)^2 (1 - corr < S_j, S_k >)^2}}. \tag{9}$$

The only limitation of Pearson's correlation coefficient approach is that it is a linear measurement statistic and cannot capture the nonlinear patterns exhibited by stock series data. To simulate non-linear dependency, information theory-based statistical measures such as mutual information and partial mutual information are used [40, 147]. If we consider the closing price series of two stocks, $S_i(t)$ and $S_j(t)$, as discrete random variables, then the **mutual information (MI)** metric is calculated as

$$MI_t(S_i(t), S_j(t)) = H(S_i(t)) + H(S_j(t)) - H(S_i(t), S_j(t)), \tag{10}$$

where $H(S_i(t))$ and $H(S_j(t))$ are marginal entropy, and $H(S_i(t), S_j(t))$ is the combined entropy of the closing price series S_i and S_j at time instance t. It is calculated as

$$H(S_i(t)) = -\Sigma_{m=t-p+1}^{t} P(S_i(m)) \log(P(S_i(m))), \tag{11}$$

$$H(S_i(t), S_j(t)) = -\Sigma_{m=t-p+1}^{t} \Sigma_{n=t-p+1}^{t} P(S_i(m), S_j(n)) \log(P(S_i(m), S_j(n))). \tag{12}$$

Here, P represents probability. There are different ways to define this probability function, such as modelling it as a frequency measurement, as represented in work [40].

5.1.2 Corporate Relation Graph Construction Method. Various publicly available relations, such as industry-sector, consumer-supplier, and shareholding patterns, are considered when creating a corporate relation graph. Here, we can build a single-relational or multi-relational graph. These graphs can be directed or undirected, weighted or unweighted.

— **Single-relational graph:** Vertices and edges in this graph have the same relation type, r. If two vertices $< v_i, v_j >$ belong to relation r, then an edge is considered between them.
— **Multi-relational graph:** In this graph, vertices are connected with different types of relationships and have multiple edges as well. Consider the relationships: $R = \{r_1, r_2, \ldots, r_m\}$. If $< v_i, v_j >$ are part of the relationship r_k, then an edge is considered between two vertices $< v_i, v_j >$ for some relation $r_k, k \in \{1, 2, \ldots m\}$ such that $e_{ij}^{r_k} = 1$.

The adjacency matrix A is a binary matrix $\in N \times N$ with entries 0 and 1 for a single relational graph. The adjacency matrix A in a multi-relation graph is a tensor of dimension $\in N \times N \times k$, where k represents the number of underlying relations. These are pre-defined relationships; the selection of relationships is a subjective choice, and this introduces human bias. While considering multiple relationships, it becomes necessary to extract only meaningful associations.

5.1.3 Textual Graph. Textual graphs are used for sentiment analysis in stock market forecasting. Each node is associated with a news headline or tweet. An edge is constructed between two nodes if they are connected by some common news headline or article. The objective is to convert

Table 2. Description of Historical Information Encoders

Encoder	Description	Expressions
LSTM [53]	LSTM was proposed as a special form of RNN to solve the vanishing and exploding gradient problems. Through their powerful gate mechanisms, these models can relate past information.	$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f)$ $i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i)$ $o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o)$ $c\hat{e}_t = tanh(W_c x_t + V_c h_{t-1} + b_c)$ $ce_t = f_t * ce_{t-1} + i_t * c\hat{e}_t$ $h_t = o_t * tanh(ce_t)$[7]
GRU [27]	The GRU model is an optimized version of the LSTM model and has fewer parameters as it has only two gates: the update gate and the reset gate, which makes its training efficient.	$z_t = \sigma(U_z x_t + W_z h_{t-1})$ $r_t = \sigma(U_r x_t + W_r h_{t-1})$ $\tilde{h_t} = tanh(U_h x_t + W_h(r_t * h_{t-1}))$ $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h_t}$[8]
CNN [54]	CNNs apply filters to capture the hidden abstract features. Because CNN models are applicable to grid-structured data, various variants, such as 1D CNN and 2D CNN are proposed to extract historical features from time series data.	CNNs consist of convolution layers followed by pooling layers. These networks effectively capture local patterns.
TCN [9]	A Temporal Convolutional Network (TCN) combines causal and dilated convolutions to capture long-range dependencies. Causal convolutions ensure output dependence only on previous input data, while dilation mechanisms expand the receptive field to incorporate historical periods.	$h_t = \Sigma_{i=0}^{k-1} filter(i) \cdot x_{t-(d_r \cdot i)}$ $filter : \{0, 1, 2, \ldots k-1\} \rightarrow \mathbb{R}$: filters, d_r: dilation rate

text into a vector representation while preserving the semantic information. To convert this unstructured text data into vectors, various NLP approaches are used. To embed the semantics and syntax of the word and its adjacent words, word embedding techniques such as Word2Vec [100] and GloVe [111] are used. Utilizing word embedding techniques for large texts becomes tedious, and the information extracted is limited to the semantics of a word. To alleviate this problem, sentence embedding methods are proposed where the entire sentence or document and its semantics are represented as vectors. Doc2Vec [79] and Universal Sentence Encoders [18] are examples of sentence-level embedding methods. Language models assign probability to word sequences and extract the rich hidden semantics of a language by predicting the next word based on previous words. The **bidirectional encoder representation from the Transformer (BERT)** model [32] has achieved state-of-the-art performance for many NLP tasks, and the financial domain is no exception. Some researchers have utilized the BERT model to perform sentiment analysis for the financial domain [73, 123]. Moreover, the FinBERT model [5] is proposed by fine-tuning a pre-trained BERT model on financial data.

Once the vector representation is obtained using one of the above-mentioned embedding techniques, these extracted textual features, along with the market data, are given to the historical information and relational encoders. To fuse textual features with market features, most commonly concatenation [25, 31], linear combination [33, 60], or dynamic neural-network-based interaction mechanisms [43, 82, 118] are used.

5.2 Historical Information Encoder

The Historical Information Encoder updates the node embedding based on the feature tensor $F_t \in R^{N \times p \times D}$ of N stocks. The objective is to encode the historical data of window size p. The encoder produces an updated feature matrix, $F_t' \in R^{N \times U}$, where U is a node dimension following the update. In the literature, sequential models such as RNNs, LSTM [53], and GRU [27] are frequently used to process the time series data due to their ability to process the entire sequence. In addition, to process the temporal dimension, CNNs and their variants have also been used [9, 54]. Table 2 describes the working mechanisms of the most commonly used historical encoders that take $X_t \in R^{p \times D}$ as an input feature matrix of stock S with D-dimensional features at time instance t.

[7] f_t, i_t, o_t, ce_t, h_t: forget, input, output, cell, and hidden state; b_f, b_i, b_o, b_c and W_f, V_f, W_i, V_i, W_o, V_o, W_c, V_c: learnable bias and vectors; * represents element-wise multiplication.
[8] z_t, r_t: update and reset gates; U_z, W_z, U_r, W_r, U_h, W_h: learnable weight matrices.

The historical encoders presented in Table 2 generate the hidden state h_t that encodes the previous p data points. The more description about the architectural design of the historical encoders presented in Table 2 can be found in the references given against each model. Considering the volatile nature of financial time series, each previous time step or day has a varying amount of effect on the present time step. To consider this constraint, an attention mechanism is being applied to process the fluctuating nature of time series. The notion of an attention mechanism was first introduced by Bahdanau et al. [8] for machine translation tasks as an encoder-decoder architecture. It computes an attention-weighted state h_t^a for a time series of data at time step t by taking the look-back window of size p into account. The attention mechanism is summarized by the following set of equations:

$$h_t = Encoder(h_{t-1}, x_t), \tag{13}$$

$$h_t^a = \Sigma_{i=t-p+1}^{t-1} \frac{exp(W_\alpha h_i)}{\Sigma_{k=t-p+1}^{t-1} exp(W_\alpha h_k)} h_i, \tag{14}$$

where W_α is a learnable parameter. The encoder can be any model presented in Table 2 that generates sequential embedding for the previous p time steps at each time step [40, 57, 86]. The updated embeddings obtained by the historcial encoders, along with the adjacency matrix, are given to the relational encoder module. There have also been attempts to combine CNN and LSTM models to capture the short and long-term fluctuations in financial series [90, 120, 127].

5.3 Relational Encoder

The objective of this module is to exploit the spatial dependence in the graph-structured data. As inputs, an adjacency matrix A and an updated feature matrix $F_t^{'}$ of N stocks obtained from historical information encoder are provided. Various graph embedding techniques such as node2vec [47], TransR [87], and TransE [15] are used to transform defined graph structures into numerical representations. These techniques do not consider the time-evolving node features to generate embedding, resulting in the loss of rich information. To alleviate that, various GNN models have been proposed [142]. Here, we discuss only the most commonly used GNN architectures in reviewed work for stock market forecasting. Graph Neural Networks are broadly classified into three categories: recurrent graph neural networks, spatial convolution networks, and spectral convolution networks. The recurrent graph models only consider the mutual dependence of one-hop neighbours and lack the extendability to consider the cyclic mutual dependence among nodes [142]. Due to this, the RecGNN model is the least used. The dominant graph models in spatial and spectral categories are discussed next.

Spectral Convolution Graph Neural Networks: These types of networks are inspired by graph signal processing theory and work in the frequency domain. They capture the mutual dependency among nodes by considering a fixed number of layers with different weights. Various models such as Spectral-CNN [16], ChebNet [30], and **Graph Convolutional Networks (GCN)** [71] have been proposed. GCN is a first-order approximation model of ChebNet [30] and the most widely used model for stock market forecasting. The mathematical foundation of these models is summarized by Wu et al. [142]. Due to space constraints, we will discuss the foundation of GCN only. Consider a L-layer GCN network that has the capability to capture the mutual dependence of an L-hop neighborhood, i.e., a GCN with two layers can capture the local information from a two-hop neighborhood. Layer $l's$ hidden state is denoted as H^l and is expressed as

$$H^l = f(A_{norm} H^{l-1} W^{l-1}). \tag{15}$$

To eliminate instabilities, the normalized adjacency matrix A_{norm} is induced as a normalization step. $A_{norm} = D^{-1/2}(A + I)D^{-1/2}$, where D and I represent the node's diagonal degree matrix and identity matrix, respectively. At layer $l-1$, W^{l-1} is a learnable weight matrix, $f(.)$ is an activation function, and $H^0 = F_t^{'}$ obtained from historical information encoder.

The GCN network is also considered a spatial-based method in the literature. It is defined as an aggregator function that captures the neighbourhood node's feature information. The hidden state at the input layer is initialized as an updated feature vector obtained from the historical information encoder. Considering the hidden state of node v at layer l as h_v^l and its neighbourhood as $\mathbb{N}(v)$, Equation (16) is used to update it:

$$h_v^l = g\left(\Sigma_{u \in \mathbb{N}(v) \cup v} \frac{1}{\alpha_{u,v}} h_u^{l-1} W^{l-1}\right), \qquad \forall v \in V. \tag{16}$$

Here $\alpha_{u,v}$ is a normalization factor that defines explicit strength between nodes u and v defined as $\sqrt{|\mathbb{N}(v)||\mathbb{N}(u)|}$. Where $|\mathbb{N}(u)|$ is the number of one hop neighbors. $g(.)$ is an activation function, and matrix W is a learnable parameter.

With the increase in the depth of the layers, the GCN model generates equivalent embedding for all classes, leading to poor generalization capability. This problem is known as the oversmoothing problem and is caused by the graph structure's topology. To overcome this, DropEdge [61] is proposed, which eliminates redundant edges at each training epoch. Another approach [19] introduces the regularization term in the learning objective.

Spatial Convolution Graph Neural Networks: Opposed to spectral GNNs, this model directly works on the spatial domain and learns node embedding. GraphSAGE [50], **Graph Attention Network (GAT)** [131], and **Gated Attention Networks (GaAN)** [158] are the most widely used neural network models in this category. The GAT model is widely used in stock market forecasting to extract relevant relationships from pre-defined multiple relationships or to find the most influential neighbors. Inspired by the masked self-attention mechanism, GAT assigns different weights implicitly to the neighbourhood nodes. The following set of equations updates the hidden state of node v at layer l, denoted as h_v^l:

$$h_v^l = g\left(\Sigma_{u \in \mathbb{N}(v)} \alpha(u,v)^{l-1} h_u^{l-1} W_1^{l-1}\right) \qquad \forall v \in V, \tag{17}$$

$$\alpha_{(u,v)}^{l-1} = \frac{h(LeakyReLU(r^T[W_2^{l-1}h_u^{l-1}||W_2^{l-1}h_v^{l-1}]))}{\Sigma_{u \in \mathbb{N}(v)} h(LeakyReLU(r^T[W_2^{l-1}h_u^{l-1}||W_2^{l-1}h_v^{l-1}]))}. \tag{18}$$

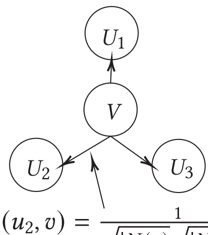
The parametric strength between two nodes at layer $l-1$ is defined by $\alpha(u,v)^{l-1}$, $g(.)$ and $h(.)$ are activation functions, and W_1, W_2, and r are learnable matrices and vectors, respectively. Figure 6 depicts the difference between GCN and GAT architectures. The GaAN [158] model extends the GAT model by introducing a multi-head self-attention mechanism.

5.4 Task-specific Layer

The relational encoder generates a high-dimensional node representation that captures the spatial dependency such that nodes that are related in the non-Euclidean domain will have similar embedding. These embeddings are fed into a **multilayer perceptron (MLP)** neural network to perform task-specific operations. Based on the underlying task, various objective functions are defined to train a model. The most commonly used loss functions are discussed in Table 3.

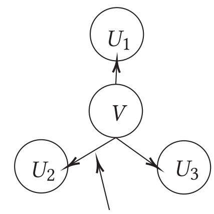
6 Prediction Models

In this section, we review the graph-based stock prediction models and present their approaches to define a financial knowledge graph and their respective modeling techniques. Moreover, the features, dataset, evaluation parameters, baseline models, and duration considered for prediction



$\alpha(u_2, v) = \frac{1}{\sqrt{|N(v)}\sqrt{|N(u_2)|}}$

(a) GCN.



$\alpha(u_2, v) = softmax_{N(v)}[r[Wh_{u_2}||WH_v]]$

(b) GAT.

Fig. 6. Graph Neural Network: (a) GCN: The strength between nodes is updated explicitly. (b) GAT: The strength between nodes is parametric function based on hidden representations.

Table 3. Description of Loss Functions Used for the Respective Tasks

Task	Description	Loss function
Classification	The classification task predicts the price movement of a stock classified into one of the c classes (Up, Down, or Neutral). Cross-entropy loss is most commonly used to train a classification model that calculates the difference between two probability distributions.	$\mathbb{L}_{Classification} = \Sigma_{i=1}^{N}\Sigma_{j=1}^{c} y_{ij} \ln \hat{y_{ij}}$ N: Number of stocks. c: Number of classes. y and $\hat{y}$: Actual and predicted class.
Regression	The regression task predicts the future closing price of a stock. The loss function calculates the difference between the actual and predicted values.	$\mathbb{L}_{MSE} = \frac{\Sigma_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert^2}{N}$ $\mathbb{L}_{MAE} = \frac{\Sigma_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert}{N}$ $\mathbb{L}_{RMSE} = \sqrt{\frac{\Sigma_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert^2}{N}}$ N: Number of stocks. y and $\hat{y}$: Actual and predicted closing prices.
Stock recommendation	The recommendation model identifies the most profitable stocks based on the return ratio ($return$) defined in Section 4.1. To train such a model, pairwise ranking loss is combined with regression loss. The first term in this equation computes the regression loss, and the second term defines the pairwise ranking loss.	$\mathbb{L}_{Recommend} = \mathbb{L}_{regreesion} + \alpha\mathbb{L}_{ranking}$ $\mathbb{L}_{regression} : \lvert return - ret\hat{u}rn\rvert^2$ $\mathbb{L}_{ranking} : \Sigma_{i=0}^{N}\Sigma_{j=0}^{N} max(0, -\beta)$. $ret\hat{u}rn$: Predicted return. α: Balancing hyperparameter. $\beta : (ret\hat{u}rn_i - ret\hat{u}rn_j)(return_i - return_j)$

are highlighted for each model. As discussed, we have classified various models according to the underlying tasks they perform, i.e., classification, regression, and recommendation. The classification task aims to predict the next day's price movement, i.e., up, down, or neutral. A regression task predicts the next day's closing price of an underlying stock. Stock recommendation models generate a ranked list of highly profitable stocks. In subsequent sections, all graph-based models for each task are discussed in detail.

6.1 Classification Models

Ding et al. [33] proposed a **Knowledge Graph Event Embedding (KGEB)** model that incorporated knowledge graphs into event embedding to enhance the predictive performance on the S&P 500 dataset. To obtain event and knowledge graph embedding, a **neural tensor network (NTN)** is used. A joint event embedding objective function is used to train a model. By incorporating relationship-driven knowledge, semantically meaningful event embedding is obtained, as it provides external knowledge about the event tuples. Extending the work done in Reference [33], Deng et al. [31] proposed a KDTCN model that additionally incorporated historical price data to predict the stock movement of stocks listed in the DJIA index. Moreover, to model historical features, TCN was used. Liu et al. [88] considered the news data, correlation between stocks, and quantitative features to predict the rise and fall of a stock. To build a knowledge graph, different pre-defined relationships were considered, and from news articles and annual reports, correlated companies were extracted. Using the TransR model [87], node embedding is obtained, and Euclidean distance is used to build an adjacency matrix. A correlation matrix, along with a sentiment vector and quantitative features, was given to the GRU model to make final predictions. Relational modeling techniques based on TransR still fail to extract semantic relationships between entities and require structured data.

Chen et al. [24] proposed a prediction pipeline and joint prediction model based on GCN by considering shareholding ratio to construct a static and weighted graph for stock price movement prediction for 2,988 companies listed on the Chinese stock market. A prediction pipeline model updated the target company's representation via the Node2Vec model, and cosine similarity is used to find the top N relevant companies. whereas the joint prediction model used the LSTM model to extract temporal dependence and the GCN model to update node embedding. The proposed joint LSTM and GCN model outperformed the pipeline prediction model as well as the individual LSTM and GCN models. Li et al. [81] proposed a LSTM-RGCN model to study the effect of the financial news on the overnight movement of stocks listed on the Tokyo Stock Exchange. Overnight news headlines are associated with each node that represents a particular company, and associations between nodes are defined using a correlation matrix calculated from the historical market price. By building a relational graph, predictions on stocks are made that are not directly associated with any news. LSTM is used to address the oversmoothing problem of the GCN model. But the proposed framework focused only on a single relationship between stocks to build a graph.

Considering the dependency of stock data on multiple relationships Kim et al. [70] proposed the HATS model to investigate the impact of multiple relationships on the price movement of S&P 500 index stocks. They utilized the hierarchical attention mechanism to extract rich and meaningful relationships for the target company. To verify the effectiveness of the model classification, profitability measures were considered. The proposed model predicted the future movement of an individual company as well as the S&P 500 index. To further modify this work, an extended model is proposed by incorporating textual features for the S&P 500 and CSI 300 indexes [60]. Due to the scarcity of textual data, the inclusion of news data did not improve performance in terms of the Sharpe Ratio, a profitability measure. Alshara et al. [3] proposed a multi-layer graph attention-based FHAN model that followed the same approach as defined in Reference [70]. As a novelty, the Fraudar Algorithm is introduced to remove the highly suspicious nodes as they induce noise. Although the experiments on individual stock price movement prediction on S&P 500 listed stocks showed little improvement with the proposed Fraudar algorithm. Sawhney et al. [118] proposed the MAN-SF model, which used bilinear transformation to combine historical data and tweets. To capture the temporal dependency, GRU is proposed. GAT is applied to pre-defined WikiData relations to predict the price movement of stocks in the S&P 500. This outperformed the [70] model in terms of accuracy. Sun et al. [125] extended the work of Reference [70] by incorporating WikiData graphs as well as correlation and mutual information graphs for the S&P 500 index. They applied a multi-channel GCN model to combine these three different graphs, and the GRU model was utilized to encode the historical sequence.

Jiexia et al. [68] proposed the Multi-GCGRU model to predict the stock price movement of companies listed on the CSI 300 and CSI 500 index and incorporated multiple predefined shareholding, industry, and topicality graphs. Furthermore, GRU was utilized to model the temporal dependence present in financial time series data. Moreover, a dynamic GCGRU model is also proposed to alleviate the requirement of prior knowledge to construct a graph. Although the Multi-GCGRU with pre-defined relations outperformed the other models. Wang et al. [135] proposed a HATR (Hierarchical Attention Temporal Relational) model to extract the short and long-term volatility features and multi-graph diffusion convolutions to extract the semantic information from multiple graphs constructed from pre-defined relations. To encode historical features, a dilated causal convolution layer with Hawkes's process [78] is introduced to learn significant temporal points. Relational modeling is performed utilizing industry, topicality, and adaptive graphs. This work highlighted the fact that, apart from sequence models, convolutional neural networks can also be utilized as historical information encoder. To improve on the HATR framework [135], the HATR-I model [136] proposed node- and semantic-level graph attention mechanisms.

Li et al. [82] defined a unique way to aggregate the multi-source heterogeneous data, i.e., stock trading data, news, and graphical indicators, and proposed the Msub-GNN model to predict the future stock price trend of the Shanghai Stock Exchange, the Shenzhen Stock Exchange, and the CSI 300. Furthermore, a subgraph convolution aggregation mechanism is used to combine the multi-source heterogeneous graphs. Xu et al. [144] proposed the SK-GCN model that generated a heterogenous graph consisting of stock nodes, word nodes, and external nodes. The external nodes utilized the pre-defined industry-sector relationship to connect two stock nodes. A GCN was utilized to extract the embeddings and predict the future price movement of the considered 666 Chinese stocks. Specifically, Msub-GNN [82] and SK-GCN [144] focused on aggregating heterogeneous data types effectively, whereas the other approaches [3, 68, 70, 125, 135] focused on combining multiple pre-defined relationships only.

The selection of pre-defined relations to build a knowledge market graph requires domain expertise. To alleviate the need for pre-defined relationships, Hou et al. [55] proposed a ST-Trader framework that constructed the financial graph from the latent features learnt from the **Variational Auto Encoder (VAE)**. To encode spatial and temporal patterns, GCN and LSTM models are used, respectively, to predict the future price movement of the S&P 500 stocks on a minute dataset. Cheng and Li [25] proposed a dynamic relation graph-based AD-GAT (Attribute Driven Graph Attention Network) model. To study the momentum spillover effect, an attribute-based aggregator is designed, and a tensor fusion module is defined to combine firm-specific historical and textual data. Chen et al. [23] proposed a GC-CNN model to forecast the future trend of six stocks chosen at random from the Chinese market. To apply CNN, they transformed stock data into images and applied Dual-CNN to process market data and technical indicators. Furthermore, it utilized the **Improved Graph Convolutional Network (IGCN)** module to capture the overall stock market information. A spearman's correlation coefficient was used to measure the dynamic strength between stock pairs.

Furthermore, Chen and Robert [21] proposed the MGRN model, which considered multiple relationships as well as news data to forecast the direction of the STOXX Europe 600 index. Unlike the MAN-SF model [118], they used multiple graphs to enhance the prediction accuracy. The node embedding extracted from the news served as a node feature in multiple graphs. At last, the RNN model is used to consider the temporal pattern of news. It concluded that considering multiple relationships at the same time improved the prediction performance. Another approach proposed by Zhnag et al. [162] considered technical indicators, textual data, and a pre-defined market graph and proposed DGATS to predict the future price movement of the NASDAQ and NYSE stocks. The Kronecker Tensor Fusion was used to fuse the technical and textual data. They utilized the LSTM with temporal attention to generate the sequential embedding and the attribute driven Graph Attention Network to learn the dynamic relationship strength to generate relational embedding from the defined stock market graph. They utilized industry category, supply chain, competition, customer, and strategic alliance to build the graph.

A bi-typed dual attention graph model is proposed in Reference [164] to learn the pre-defined explicit as well as implicit relationships existing between companies. To build a knowledge market graph, bi-typed entities that are a set of companies and their associative executives are considered, which contain rich information about listed companies. To model the relational dimension, an inter and intra-class dual attention mechanism is incorporated. In contrast to previous approaches, here a heterogeneous graph is considered to predict the stock market direction of the CSI300 and CSI100.

Jafari and Haratizadeh [63] proposed a GCNET model that constructed a dynamic influence graph from historical and technical indicators to study the latent association between the stock pairs for the NASDAQ index. A statistical classifier **Quadratic discriminant analysis (QDA)** is used to train prediction models that define the influence between stock pairs. To improve the predictive power, the model was trained in a semi-supervised setting. To assign plausible initial labels, a novel **Plausible Label Discovery (PLD)** framework is proposed, and the final prediction for all stocks is made using the GCN network. To encode historical information, a set of influence graphs is constructed and processed. This approach defined a unique way to build a network based on historical data. Tian et al. [129] proposed **Graph Evolution Recurrent Unit (GERU)** to learn the evolving dependencies from historical features in a dynamic manner. The clustered ADGL module is utilized to learn the complex interdependencies, and the **Gated Graph Recurrent Unit (GGRU)** is proposed to extract the temporal features. The classification models are summarized in Table 4.

6.2 Regression Models

Yin et al. [153] utilized the GCN network to extract features for a particular company that embedded the price of similar companies. Furthermore, the GRU model was incorporated to model the temporal dependence. To train a model, multi-task learning is utilized, where each task learns its own unique GRU parameters but shares a common GCN network to enhance the model's performance. Nagendra and Haritha [104] proved that there exists a correlation between various sector indices and the NIFTY index of the Indian stock market. This study also highlighted the fact that there is some kind of dependency between sectors and companies as well. Based on this, Yash et al. [151] proposed the GCN model to predict the next day's open, low, close, and high prices of 87 companies listed on the BSE100 index of the Indian stock market. To build a graph, each company's vector representation was obtained from the average stock price. Furthermore, the outcome of a weighted linear combination of three different correlation measures computed from the vector representation of each node was used as an edge weight. Sun et al. [125] and Hou et al. [55] proposed the MCT-GCN network and ST-Trader framework to perform the classification as well as the regression task, with the same settings discussed in Section 6.1.

Tao et al. [127] proposed a framework to capture the mutation points causing abrupt changes. To accommodate the mutation points, a piece wise loss function is defined for different intervals to make the model training efficient. The prediction layer is given feature vectors of related companies and mutation points to predict the price of the target stock. This work highlighted the fact that to encode temporal dependency more efficiently, we need to consider some kind of nonlinear attention mechanism. The pre-defined WikiData relations were considered to construct a graph for the **Shenzhen Stock Exchange (SZSE)**.

To learn the dynamic strength of static predefined relations, Matsunga et al. [98] used the same approach as proposed in Reference [39] for the Nikkei 225 index. As a minor adjustment, they trained the model in an end-to-end manner to increase the training efficiency. Moreover, the rolling window analysis approach was used to divide the dataset. Here, a comprehensive knowledge graph was built by also considering the companies not listed on the Nikkei 225 index that have various corporate relationships. Xu et al. [146] proposed a **Relational Event-driven Stock Trend (REST)** forecasting framework that considered news events and their cross-effect on related stocks. Event information encoders utilized multi-head attention to encode the different events and their feedback, which is reflected in historical market data. Following this, the stock context encoder used the GCN layer to study the dynamic cross-correlation of events on correlated stocks to make predictions. In conclusion, they have defined a unique way to fuse event information and historical data. GALSTM [152], another model, used the Multi-dimensional Hawkes process to extract the

Table 4. Review on Classification Models

Model	Features	Dataset	Evaluation Parameter	Baseline Models	Duration
KGEB (2016) [33]	News headlines	S&P 500	Accuracy, MCC	WB-CNN	Oct 2006–Nov 2013
KDTCN (2019) [31]	Historical data, News headlines	DJIA	Accuracy, F-score	ARIMA, LSTM, CNN, TCN, KDEB-TCN	08/08/2008–01/01/2016
[88] (2019)	News, Fundamental data	Chinese A-share	Accuracy	Different versions of own model	January 2017–May 2018
LSTM+GCN (2018) [24]	Open, Low, High, Close, Volume	China stock market	Accuracy	LR, LSTM, GCN, node2vec+LSTM, LINE+LSTM, DeepWalk+LSTM	29/04/2017-31/12/2017
LSTM-RGCN (2020) [81]	News Headlines	TPX-100, TPX-500	Accuracy	Random Forest, Naive Bayes, HAN, S-LSTM, Transformer	01/01/2013–09/28/2018
HATS (2019) [70]	Price change rate	S&P 500	Accuracy, F-score, Return, Sharp ratio	MLP, CNN, LSTM, GCN, GCN top 20, TCN	08/02/2013–17/06/2019
ML-GAT (2022) [60]	Rate of price change, Textual data	S&P 500, CSI 300	Accuracy, F-score, Average return, Sharp ratio	MLP, CNN, LSTM, TGC, GCN	–
FHAN (2022) [3]	Logarithmic rate of return	S&P 500	Accuracy, Precision, Recall, F-score	CNN, LSTM, GCN, MLP, LSTM-attention	–
MAN-SF (2020) [118]	Close, High, Low, Tweets	S&P500	Accuracy, F-score, MCC, Sharp ratio	Random model, ARIMA, LSTM, RF, TSLDA, HAN, StockNET, HATS, LSTM-GCN	01/01/2014–01/01/2016
MCT-GCN (2020) [125]	Historical prices	S&P 500	RMSE, MAE	LSTM, T-GCN	08/02/2013–17/06/2019
Multi-GCGRU (2020) [68]	Open, High, Low, Close, Volume	CSI 300, CSI 500	Accuracy, Precision, Recall, F-score, MCC	ARIMA, SVM, RF, ANN, LSTM, different versions of own model	June 2015–December 2019
HATR (2021) [135]	Historical data	S&P 500, CSI 300,TPX 100	Accuracy, AUC, F-score, MCC	SVM, RF, DA-RNN, SFM, GCN, TPA-LSTM, TGC, HMG-TF, Inception time, HATR-MT, HATR-MR	2016–2020
HATR-I (2022) [136]	Historical data	S&P 500, CSI-300, TPX-100	Accuracy, AUC, F-score, MCC	SVM, RF, DA-RNN, SFM, TPA-LSTM, Inception Time, HMG-TF, GCN, TGC, TGAT, HATR	July 2015–February 2020
Msub-GNN (2022) [82]	Market data, News, Technical data	Shanghai, Shenzhen, CSI 300	Acuracy, F-score	SVM, MKNN, RF, LSTM, TeSia	11/01/2013–25/11/2019
SK-GCN (2023) [144]	Market data	Chinese stock market	Accuracy, F-score	Logistic regression, SVM, NB	–
ST-Trader (2021) [55]	Market data, Fundamental data	S&P 500	Accuracy, Precision, Recall, F-score	LR, FCNN, LSTM	Jan 2010–Dec 2010
AD-GAT (2021) [25]	Technical indicators, Textual data	S&P 500	DA, AUC	LSTM, GRU, eLSTM, GCN, GAT, TGC	08/02/2011–18/11/2013
GC-CNN (2021) [23]	Market data, Technical Indicators	Six stocks from chinese market	Accuracy, Precision, Recall, F-score, Sharp ratio, AAR	Dual-CNN, CNN-TA, CNN-LSTM, GAF-CNN, SVM, MLP, DT	2010–2020
MGRN (2022) [21]	Close	STOXX 600	Accuracy, Sharp ratio	Random model, ARIMA, Mean-BERT, MAN-SF	–
[164] (2022)	Historical data, Sentiment data	CSI100, CSI300	Accuracy, AUC, IRR, Sharp ratio	LSTM, GRU, GCN, GAT, RGCN, HGT, MAN-SF, STHAN-SR	21/11/2017–31/12/2019
DGATS (2023) [162]	Market data, Technical data, Textual data	NASDAQ, NYSE	Accuracy, AUC, F-score, MCC	StockNet, LSTM+PS, LSTM+GCN, RSR, AD-GAT, TRAN, RA-AGAT, MAC, TRPCA	01/02/2013–12/08/2017
GCNET (2022) [63]	Historical data, Technical indicators	NASDAQ	Accuracy, MCC	ALSTM, StockNET, HATS, CNN-pred, Adv-LSTM, ExLift+DiMexRank, SACLSTM, Price graphs	01/01/2011–01/01/2021
GERU (2023) [129]	Historical data	S&P 500, CSI 100, CSI 300, Russell 1K, Russell 3K	Accuracy, Precision	LST, ALSTM, Adv LSTM, CSG-GCN, DSG-GCN, HATS, FinGAT, HAD-GNN, MTGNN, AGCRN, SDGNN	–

correlation between stocks. Given a correlation matrix, a graph attention layer is built to update the node features based on the spatial dimension. Afterwards, the LSTM model is used to make predictions for high-frequency data.

Wang et al. [134] proposed the MG-Conv model to predict the 42 stock indices of the Chinese market. They have utilized the 1D convolution network first to extract the deep features, and these features are fed into graph neural networks. As typical graph construction methods cannot be applied to build a graph for an index prediction, static and dynamic graphs were built. To build a static graph, the weights of a stock in respective indices were considered, and to remove dependence on pre-defined relations, a dynamic graph was built from historical data during the training process, treating it as a learnable parameter. Further modifications can be made to combine these graphs

Table 5. Review on Regression Models

Model	Features	Dataset	Evaluation Parameter	Baseline Models	Duration
GC-GRU (2021) [153]	Median share price	DJIA (43 stocks), ETFs	RMSE, MAE, R^2, Accuracy forecast	GRU	22/11/2010–05/07/2018
[151] (2021)	Open, Low, High, Close	87 stocks from BSE100	RMSE	–	01/01/2018–10/10/2020
MCT-GCN (2020) [125]	Historical prices	S&P 500	MAE, RMSE	LSTM, T-GCN	08/02/2013–17/06/2019
[127] (2022)	Closing price	Shenzhen	MAE, MSE	LSTM, GRU, CNN, DA-RNN, GAN, WGAN, CNN-LSTM, BiLSTM-M, BiLSTM-S, ConvLSTM	04/01/2010–31/12/2019
[98] (2019)	Close and 5,10,20,30 days moving average	Nikkei index	Return ratio, Sharp ratio	LSTM, same model with different relationships	–
REST (2021) [146]	Market data, Textual data	CSI 300,CSI 500	RMSE, MAE, MedAE	ARIMA, TGCN, HAN, GCN, RGCN	2013–2018
GALSTM (2021) [152]	Trade price	China A share	RMSE	ARIMA, LSTM, Seq2seq	–
MG-Conv (2022) [134]	Open, Low, Close, High, Volume	42 indices from China market	MAE, MSE	ANN, LSTM, 3D-CNN, CNN-LSTM, GC-CNN, AD-GAT	2009–2021
VGC-GAN (2023) [91]	Market data	ETF, Shanghai Stock Exchange, DJIA	MAE, RMSE, MAPE, and Accuracy forecast	GRU, GCGRU, SCINet, StemGNN, PF-LSTM, DLinear, FEDformer, N-hits, WaveForM, PatchTST	2010–2020

using a non-linear approximation. Ma et al. [91] proposed the VGC-GAN model that considered multiple correlation graphs built from historical market data. The **Generative Adversarial Network (GAN)** was proposed to enhance prediction performance. The multi-GCN and GRU models were utilized as generators, and CNN as a discriminator. To reduce the noise, the **Variational Mode Decomposition (VMD)** technique is used to generate the subsequences, which are then given to the generator model. The regression models are summarized in Table 5.

6.3 Stock Recommendation Models

Feng et al. [39] proposed the **Relational Stock Ranking (RSR)** model by designing the temporal graph convolution module to select the stock list with the highest expected revenue for the NYSE and NASDAQ stock markets. Furthermore, a time-sensitive relation strength function was proposed to propagate the time-aware embedding that considered sector-industry, supplier-consumer, and ownership relations into the model. Gao et al. [43] extended the RSR model [39] by incorporating textual data and proposed the **Time-aware Relational Attention Network (TRAN)** to generate the stock recommendation list for the NASDAQ and NYSE markets based on a higher return ratio. They introduced the TRA (time-aware relational attention) unit, similar to RSR, to model dynamically changing relationship strength.

To identify hidden concepts apart from pre-defined relationships, Xu et al. [145] proposed a HIST model to learn the dynamic relevance of stocks having pre-defined relationships as well as hidden concepts. The embedding of pre-defined and hidden concepts is updated at each step. It outperformed baseline models on the CSI 300 and CSI 100 indices. Yanshen et al. [150] improvised a model [145] and proposed a data-driven graph model (SDGNN) to learn static and dynamic graph structures for the stock recommendation task for the CSI300 and CSI100 datasets. A static graph is learned during the training process, and a dynamic graph is built from historical features extracted from the GRU model. A graph interaction module is defined to fuse the information between two graphs. For static and dynamic graphs, a graph convolution layer is designed.

Hsu et al. [57] proposed the **Financial Graph Attention Network (FinGAT)** model, which applied an attention-based GRU model to learn the short and long-term fluctuations in the stock time series data. To alleviate the need to describe the pre-defined relationships, a fully connected graph

Table 6. Review on Stock Recommendation Models

Model	Features	Dataset	Evaluation Parameter	Baseline Models	Duration
RSR (2019) [39]	Close and 5,10,20,30 days moving average	NYSE, NASDAQ	MSE, MRR, IRR	SFM, LSTM, Rank LSTM, GBR, GCN	01/02/2013–12/08/2017
TRAN (2020) [43]	Close, High, Open, Low, Volume and Textual data	NYSE, NASDAQ	MSE, MRR, IRR	LSTM, SFM, Rank LSTM, GCN, GAT, RSR-E, RSR-I, TRAN(Doc2vec)	01/02/2013–12/08/2017
HIST (2021) [145]	Open, High, Low, Close, Volume, Volume weighted average price	CSI 300, CSI 100	IC, RankIC	MLP, LSTM, GRU, SFM, GAT, ALSTM, Transformer, ALSTM+TRA	01/01/2007–31/12/2020
SDGNN (2022) [150]	Open, High, Low, Close, Volume, Volume weighted average price	CSI 300,CSI 100	IC, RankIC	MLP, SFM, LSTM, GRU, ALSTM, ALSTM+TRA, GATs, HIST	01/01/2007–31/12/2020
FinGAT (2021) [57]	Open, Close, High, Low, Moving averages	Taiwan, S&P 500, NASDAQ	MRR, Precision, Accuracy	MLP, GRU, GRU-attention, FineNet, RankLSTM	—
RA-AGAT (2021) [40]	Open, High, Low, Close, Volume	SSE	CRoI, MSE, MRR	LR, ARIMA, SFM, Attn-LSTM, Attn-LSTM-GCN, Attn-LSTM-GAT	01/01/2009–12/07/2019
STHAN-SR (2021) [116]	Historical data, Return ratio, 5, 10, 20 and 30 day Moving Average	NASDAQ, NYSE, TSE	Sharpe ratio, IRR	ARIMA, ALSTM, HGCluster, GCN, HATS, DQN, RSR	Jan 2013–Aug 2020
RT-GCN (2023) [165]	Market data	NASDAQ, NYSE, CSI	MRR, IRR	ARIMA, A-LSTM, SFM, LSTM, DQN, iRDPG, Rank_LSTM, RSR, RT-GAT, STHAN-SR	02/03/2020–31/12/2020
HyperStockGAT (2021) [117]	Return ratio,	NASDAQ, NYSE, TOPIX 100, Chinese A share market	Sharpe ratio, IRR, NDCG	SFM, LSTM, ARIMA, A-LSTM, GCN, HATS, DQN, iRDPG, Rank LSTM, RSR E, RSR I	2013–2020

is considered between sectors, and a spatial GAT is applied to extract the embedding. To train a model, a multi-task learning objective is considered for Taiwan, S&P 500, and NASDAQ stocks. Feng et al. [40] proposed the **Relation-aware Dynamic Attributed Graph Attention Network (RA-AGAT)** model to generate recommendation lists of the top-n high return ratio stocks of the Chinese A-share market. To build a graph, four different correlation measures, i.e., the Pearson coefficient, the partial correlation coefficient, the **Detrended Cross Correlation Analysis (DCCA)** coefficient, and mutual information have been explored. Among them, the mutual information measure has given the best performance. Moreover, the attention network is used to selectively aggregate the neighbour features of the attributed graph by utilizing edge information. Moreover, loss of correlation is introduced to train a model.

Sawhney et al. [116] proposed the STHAN-SR model to recommend the most profitable stocks based on return ratio. To encode historical features, LSTM is used, and a hypergraph is constructed from pre-defined industry and WikiData relations. In contrast to normal graphs, a hypergraph connects a set of vertices instead of two vertices. To encode relational dependency, attention-based hypergraph convolution is proposed. Zheng et al. [165] proposed the RT-GCN model to extract the relational and temporal features for each stock. They have built a relation-temporal graph from the pre-defined WikiData relations and connections established by market features at different time steps. The GCN and three proposed relation aware strategies are applied to extract the features of a stock and given to the **Temporal Convolutional Network (TCN)** to predict the ranking scores of the stocks. It has outperformed the STHAN-SR model [116]. Sawhney et al. [117] proposed the HyperStockGAT model to recommend top-ranking stocks listed on the NASDAQ, NYSE, TOPIX 100, and Chinese A share markets. The attentive temporal convolution is used to extract the temporal patterns, followed by the hyperbolic graph convolution layer to embed the interdependencies between the stocks. A financial knowledge graph is built using pre-defined WikiData relationships. Table 6 summarizes stock recommendation models.

All graph-based models discussed above are illustrated in Figure 7 in a chronological manner and also indicate the category of task they perform.

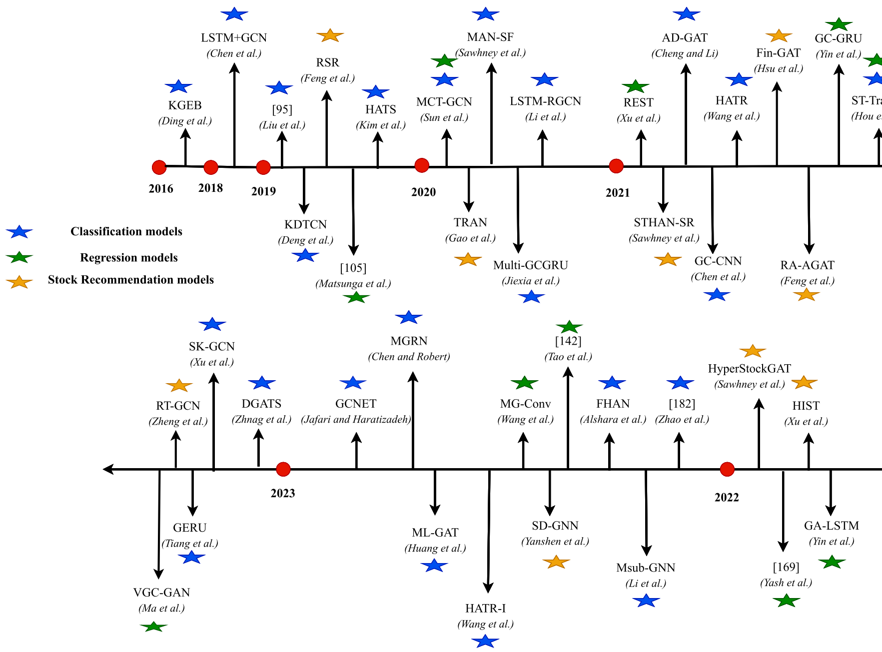


Fig. 7. Chronological summarization of graph models for stock market forecasting.

7 Analysis and Discussion

In this section, we analyze the results from each article in terms of various modules and evaluation parameters.

7.1 Analysis on Modules

The analysis of models based on the different modules discussed in Section 5 is represented here. Table 7 represents the analysis of graph construction technique, historical encoder, and relational encoder in detail.

In the next section, we draw some conclusions based on the analysis done.

7.1.1 Discussion on Modules.

— **Remarks on Graph Construction Module:**

(1) Most of the reviewed work has considered pre-defined static relationships to build a graph [3, 21, 24, 31, 33, 60, 68, 70, 88, 116–118, 125, 134–136, 144, 165]. This requires the efforts of the domain expert.

(2) The work cited in References [39, 43, 57, 98, 145, 146, 162] considered pre-defined relationships to learn time-evolving relationship strength, but the strength between stock pairs is learned dynamically.

(3) To remove human intervention to define significant relations, various statistical measures calculated from the market data, such as Pearson's correlation coefficient and Sperasmen's correlation coefficient, are utilized [21, 81, 91, 151, 153]. But the strength remains static throughout the model.

(4) The ST-Trader framework [55] has considered a static adjacency matrix from the latent features learnt by the VAE model.

Table 7. Analysis on Modules

Models	Graph Construction Module	Static or Dynamic relationship	Historical Information Encoder	Relational Module
KGEB [33]	YAGO relations	Static	–	Neural Tensor Network
[88]	Corporate, Shareholding, Supplier-consumer, Management	Static	GRU	TransR
KDTCN [31]	Wikdata relations	Static	TCN	TransE
STHAN-SR [116]	Industry, WikiData relations	Static	LSTM	Hypergraph Attention
LSTM-GCN [24]	Shareholding ratio	Static	LSTM	GCN
HATS [70]	WikiData multiple relations	Static	LSTM	Hierarchical Graph Attention
MCT-GCN [125]	WikiData relations, MI, Correlation matrix	Static	GRU	GCN
LSTM-RGCN [81]	Correlation matrix	Static	LSTM	GCN
MAN-SF [118]	WikiData	Static	GRU	GAT
MGRN [21]	Correlation, Industry, Supply chain graph	Static	RNN	GCN
HATR-I [136]	Industry, Topicality, Shareholding graph	Static	Dilated causal convolution	GAT
Multi-GCGRU [68]	Shareholding, Industry, Topicality	Static	GRU	GCN
FHAN [3]	WikiData multiple relations	Static	LSTM	Hierarchical Graph Attention
Msub-GNN [82]	Textual data and Technical Indicator	Static	LSTM	GCN
GC-GRU [153]	Correlation matrix	Static	GRU	GCN
VGC-GAN [91]	Multiple correlation matrix	Static	GRU, CNN	GCN
[151]	Correlation matrix	Static	–	GCN
ML-GAT [60]	WikiData multiple relations	Static	LSTM, BERT	Hierarchical Graph Attention
[127]	WikiData multiple relations	Static	ConvLSTM	Node2vec
SK-GCN [144]	Industry-sector, textual data	Static	–	GCN
HyperStockGAT [117]	WikiData relations	Static	Attentive temporal convolution	Hyperbolic Graph Convolution
RT-GCN [165]	WikiData relations	Static	Temporal Convolution Network	GCN
ST-Trader [55]	Latent features learnt from VAE	Static	LSTM	GCN
REST [146]	Industry, Shareholding, Business, Upstream and Downstream	Static but learning dynamic strength	LSTM	GCN
RSR [39]	WikiData multiple relations	Static but learning dynamic strength	LSTM	Temporal Graph Convolution
[98]	WikiData multiple relations	Static but learning dynamic strength	RNN	GCN
TRAN [43]	Sector-industry	Static but learning dynamic strength	LSTM	Time sensitive module
Fin-GAT [57]	Sector-industry	Static but learning dynamic strength	Attention-GRU	GAT
HIST [145]	Industry, Main business	Static but learning dynamic strength	GRU	cosine similarity
DGATS [162]	Industry category, supply chain, competition, customer, and strategic alliance	Static but learning dynamic strength	LSTM with temporal attention	GAT
RA-AGAT [40]	Mutual Information	Dynamic	Attention-LSTM	Multi head GAT
AD-GAT [25]	Historical data graph	Dynamic	RNN	GAT with modification
GCNET [63]	Historical data	Dynamic	–	GCN
GALSTM [152]	Correlation matrix using Hawkes process	Dynamic	LSTM	Graph Attention layer
GC-CNN [23]	Spearman's correlation coefficient	Dynamic	Dual CNN	IGCN
GERU [129]	Historical data	Dynamic	GGRU	Clusterd ADGL
MG-Conv [134]	Weight of constituent stocks and historical data	Static and dynamic	CNN	GCN
HATR [135]	Industry, Topicality	Static and dynamic	Dilated causal convolution	GCN
SDGNN [150]	Data driven approach	Static and dynamic	GRU	GCN
[164]	WikiData and Implicit relations	Static and dynamic	GRU	Dual Attention

(5) Whereas other approaches [23, 25, 40, 63, 129, 152] have overcome the dependency on pre-defined static relations by applying a data-driven approach that can model dynamic patterns.
(6) References [134, 135, 150, 164] have incorporated static as well as dynamic relationships.

— **Remarks on Historical Information Encoder:**
(1) Most of the work has considered sequence models such as LSTM [3, 24, 25, 39, 43, 55, 70, 81, 82, 116, 146, 152], GRU [68, 88, 91, 118, 125, 145, 150, 153, 164], and RNN [21, 25, 98] to encode historical features.

(2) To capture historical features, there is a need to model short term as well as long term fluctuations. An attention mechanism can fulfill this requirement. It is adopted in the work [40, 57, 117, 162].
(3) Apart from sequence models, Convolutional Neural Networks and their variants are also used [23, 31, 91, 134–136, 165].
(4) Reference [127] has utilized a hybrid ConvLSTM model as a historical encoder.
(5) Reference [129] has proposed a Graph Gated Recurrent Unit that embeds GNN in the RNN cell.

— **Remarks on Relational Module:**
(1) Most of the work has considered GCN [21, 24, 55, 63, 68, 81, 82, 91, 98, 125, 134, 135, 144, 146, 150, 151, 153, 165] and its modification [23, 39, 43, 117] as relation encoders to exploit spatial dependence.
(2) To enhance the performance of GCN, some works have considered GAT [40, 57, 118, 136, 152, 162] and its modification [3, 25, 60, 70, 116, 164] that utilizes attention mechanism to learn the graph representations.
(3) Other approaches [31, 33, 88, 145] have applied node embedding modelling techniques such as Node2Vec, TransR, and TransE.
(4) A dynamic approach GERU [129] have applied the self attention mechanism to learn the relational embeddings.

7.2 Analysis on Evaluation Parameters

This section represents the quantitative analysis on classification, regression, and stock recommendation models based on the evaluation parameters used and their obtained values for the considered dataset. Table 8 summarizes the results of the classification graph models.

The observations made from Table 8 are as follows:

— For the S&P 500 dataset, the KGEB model [33] achieved the highest accuracy with a value of 66.93%, followed by the HATR-I model [136] with a value of 62.96% and ST-Trader [55] with a value of 58.70%.
— The SK-GCN [144] model has achieved an accuracy of 83.08% on the considered 666 stocks from the China.
— The HATR-I model [136] achieved the highest accuracy on CSI 300 and TPX 100, with a value of 70.54% and 67.95%, respectively, followed by the HATR [135], and ML-GAT model [60].
— For NASDAQ, DGATS [162] model outperformed the GCNET [63] on accuracy parameter.
— For CSI 100, GERU [129] has outperformed Reference [164] with accuracy value of 60.30%.
— For CSI 500, Multi-GCGRU [68] has achieved an accuracy of 58.58%. For TPX 500, LSTM-RGCN [81] achieved an accuracy of 56.14%.
— For NASDAQ and NYSE, DGATS [162] achieved an accuracy of 60.49% and 62.19%, respectively.
— For Russell 1K and Russell 3K dataset, GERU [129] achieved an accuracy of 63.60% and 60%, respectively.
— In terms of F-score, MAN-SF [118] outperformed other models on the S&P 500 dataset, Multi-GCGRU [68] on the CSI 300, and SK-GCN model [144] model on the chinese stock market.
— The HATS [70] has outperformed the MAN-SF [118], and ML-GAT [60] model with 1.99 sharpe ratio value on the S&P 500 dataset.

Table 9 summarizes the results of the regression graph models.

Table 8. Analysis of Classification Models on the Basis of Evaluation Parameters

Model	Dataset	Parameter	Value	Model	Dataset	Parameter	Value
HATR [135]	S & P 500	Accuracy	61.47	HATR-I [136]	S&P 500	Accuracy	62.96
		F-score	0.5780			F-score	0.5983
		AUC	0.5778			AUC	0.5887
		MCC	0.1303			MCC	0.1599
	CSI 300	Accuracy	67.70		CSI 300	Accuracy	70.54
		F-score	0.6259			F-score	0.6540
		AUC	0.6364			AUC	0.6825
		MCC	0.1519			MCC	0.1738
	TPX 100	Accuracy	65.78		TPX 100	Accuracy	67.95
		F-score	0.6206			F-score	0.6302
		AUC	0.5777			AUC	0.6356
		MCC	0.1086			MCC	0.1249
Multi-GCGRU [68]	CSI 300	Accuracy	57.54	ML-GAT [60]	S&P 500	Accuracy	39.31
		Precision	0.9603			F-score	0.5102
		Recall	0.5484			Average Return	0.1193
		F-score	0.6981			Sharpe ratio	1.8889
		MCC	0.2171		CSI 300	Accuracy	56.98
	CSI 500	Accuracy	58.85			F-score	0.5691
		Precision	0.9894			Average Return	0.1334
		Recall	0.5658			Sharpe ratio	1.8974
		F-score	0.7199	AD-GAT [25]	S&P 500	DA	0.5647
		MCC	0.2377			AUC	0.5894
[164]	CSI 100	Accuracy	57.75	GC-CNN [23]	China market	Accuracy	50.52–53.37
		AUC	60.78			$[Precision_{pos}, Precision_{neg}]$	[0.3615,0.5570]
		IRR	10.18			$[Recall_{pos}, Recall_{neg}]$	[0.5021,0.4893]
		Sharpe ratio	4.112			$[F-score_{pos}, F-score_{neg}]$	[0.4434,0.5361]
	CSI 300	Accuracy	55.79	HATS [70]	S&P 500	Accuracy	39.31
		AUC	59.36			F-score	0.3294
		IRR	16.97			Return ratio	0.0961
		Sharpe ratio	4.628			Sharpe ratio	1.9914
LSTM-RGCN [81]	TPX 500	Accuracy	56.14	FHAN [3]	S&P 500	Accuracy	35.53
	TPX 100	Accuracy	58.71			F-score	0.3311
ST-Trader [55]	S&P 500	Accuracy	58.70	ST-Trader [55]	S&P 500	Recall	0.665
		Precision	0.4920			F-score	0.5650
MAN-SF [118]	S&P 500	Accuracy	60.80	Msub-GNN [82]	Shanghai, Shenzhen, CSI 300	Accuracy	70.76
		F-score	0.6050			F-score	0.8241
		MCC	0.1950			MCC	0.4210
		Sharpe ratio	1.05	[88]	Chinese A share	Accuracy	59.40
MGRN [21]	STOXX 600	Accuracy	52.20	MCT-GCN [125]	S&P 500	Accuracy	39.21
		Sharpe ratio	0.94			R^2	0.6056
KGEB [33]	S&P 500	Accuracy	66.93	KDTCN [31]	DJIA Index	Accuracy	71.80% ± 0.25%
		MCC	0.5072			MCC	0.7388 ± 0.0030
LSTM-GCN [24]	China market	Accuracy	57.98	GCNET [63]	NASDAQ	Accuracy	56.71
						AUC	0.6078
DGATS [162]	NASDAQ	Accuracy	60.49	GERU [129]	S&P 500	Accuracy	65.70
		F-score	0.6011			Precision	0.6630 ± 0.1
		MCC	0.2558		CSI 100	Accuracy	60.3 ± 1.0
		AUC	0.5819			Precision	0.6040 ± 0.5
	NYSE	Accuracy	62.19		CSI 300	Accuracy	57.1 ± 1.4
		F-score	0.6155			Precision	0.57 ± 1.3
		MCC	0.2785		Russell 1K	Accuracy	63.6 ± 0.3
		AUC	0.5973			Precision	0.638 ± 0.5
SK-GCN [144]	Chinese market	Accuracy	83.08		Russell 3K	Accuracy	60 ± 1.1
		F-score	0.8303			Precision	0.596 ± 1.7

The observations made from Table 9 are as follows:

— In terms of MAE, REST [146] has achieved superior performance compared to other models on the Chinese market.
— In terms of RMSE, VGC-GAN [91] outperformed GC-GRU [153] for DJIA index.
— The REST [146] model has achieved an MAE value of 0.0227 and 0.0242 on the CSI 300 and CSI 500 datasets, respectively.
— On the S&P 500 stock market, ST-Trader [55] achieved an MAPE value of 0.0195 and an MdAPE value of 0.0739.
— On the Shenzhen market, Reference [127] achieved an MAE value of 0.4426 and a MAPE value of 0.0123.

Table 9. Analysis of Regression Models on the Basis of Evaluation Parameters

Model	Dataset	Parameter	Value	Model	Dataset	Parameter	Value
REST [146]	CSI 300	MAE	0.0227	GC-GRU [153]	DJIA index, ETFs	RMSE	4.1763
		RMSE	0.0301			MAE	2.3925
		MadAE	0.0181			R^2	0.8767
	CSI 500	MAE	0.0242			Accuracy forecast	52.16
		RMSE	0.0317	[98]	Nikkei Index	Return ratio	29.95
		MadAE	0.0178			Sharpe ratio	0.37
[127]	Shenzhen Market	MSE	[0.4341−3.4193]	MG-Conv [134]	China market	MSE	0.69
		MAE	[0.4426−1.3627]			MAE	0.67
		MAPE	[0.0123−0.0130]	[151]	BSE	RMSE	16.23 ± 0.006
MCT-GCN [125]	S&P 500	MAE	0.03306	GALSTM [152]	Chinese A share market	RMSE	0.79
		RMSE	0.05878				
VGC-GAN [91]	Shanghai Stock Exchange	MAE	0.3601	VGC-GAN [91]	DJIA	MAE	3.3558
		RMSE	0.4380			RMSE	4.0661
		MAPE	2.4150			MAPE	2.285
		Accuracy forecast	0.5673			Accuracy forecast	0.5712
VGC-GAN [91]	ETF	MAE	4.9362	VGC-GAN [91]	ETF	MAPE	5.865
		RMSE	5.9776			Accuracy forecast	0.5285
ST-Trader [55]	S&P 500	MAPE	0.0195	ST-Trader [55]	S&P 500	MdAPE	0.0739

Table 10. Analysis of Stock Recommendation Models on the Basis of Evaluation Parameters

Model	Dataset	Parameter	Value	Model	Dataset	Parameter	Value
Fin-GAT [57]	S&P 500	MRR	0.5687	STHAN-SR [116]	NASDAQ	Sharpe ratio	1.42
		Precision	0.0989			IRR	0.44
		Accuracy	54.25		NYSE	Sharpe ratio	1.12
	Taiwan	MRR	1.4106			IRR	0.33
		Precision	0.2756		TSE	Sharpe ratio	1.19
		Accuracy	56.82			IRR	0.62
	NASDAQ	MRR	2.03e-02	RA-AGAT [40]	SSE	MSE	2.62e-3±2.32e-3
		Precision	4.63e-03			CRoI	0.36 ± 0.17
		Accuracy	25.79			MRR	4.71e-2±2.3
RSR [39]	NASDAQ	MSE	3.79e-4±6.60e-7	TRAN [43]	NASDAQ	MSE	3.79e-4±3.90e-7
		MRR	4.09e-2±5.18e-3			MRR	3.81e-2±4.37e-3
		IRR	1.19±0.55			IRR	0.92±0.25
	NYSE	MSE	2.26e-4±1.37e-6		NYSE	MSE	2.26e-4±2.30e-7
		MRR	4.58e-2±5.55e-3			MRR	4.91e-2±4.82e-3
		IRR	0.79±0.3			IRR	1.38±0.35
HIST [145]	CSI 300	IC	0.131	SDGNN [150]	CSI 300	IC	0.137
		RankIC	0.126			RankIC	0.132
	CSI 100	IC	0.120		CSI 100	IC	0.126
		RankIC	0.115			RankIC	0.120
[117]	NASDAQ	Sharpe ratio	1.40	RT-GCN [165]	NASDAQ	MRR	0.061
		IRR	0.44			IRR-1	1.25
		NDCG	0.71			IRR-5	0.97
	NYSE	Sharpe ratio	1.10			IRR-10	1.03
		IRR	0.25		NYSE	MRR	0.056
HyperStockGAT		NDCG	0.90			IRR-1	0.92
	TOPIX 100	Sharpe ratio	1.20			IRR-5	1.10
		IRR	0.75			IRR-10	1.13
		NDCG	0.83		CSI	MRR	0.031
	Chinese A share market	Sharpe ratio	1.25			IRR-1	0.35
		IRR	1.05			IRR-5	0.35
		NDCG	0.89			IRR-10	0.38

— On the Shanghai stock market, VGC-GAN [91] achieved an MAE value of 0.3601 and a MAPE value of 2.4150.

Table 10 summarizes the results of the stock recommendation graph models.
The observations made from Table 10 are as follows:

— In terms of MRR, the RT-GCN model [165] has outperformed Fin-GAT [57], RSR [39], and TRAN [43] models on the NASDAQ and NYSE indices.

— In terms of IRR, the RT-GCN [165] has outperformed the HyperStockGAT [117], RSR [39], and TRAN [43] models for NASDAQ and NYSE markets.
— In terms of Sharpe ratio, STHAN-SR [116] has outperformed the HyperstockGAT [117] model for the NASDAQ and NYSE markets.
— On the CSI300 and CSI100 indexes, model SDGNN [150] outperformed HIST [145] in terms of IC and RankIC.

7.2.1 Discussion on Evaluation Parameters. In this section, we analyze and present the observations from the results obtained by the graph models for each classification, regression, and stock recommendation task for the considered dataset.

— For the classification task, accuracy and F-score are the most commonly used parameters to evaluate the model. The majority of reviewed work has considered indices of large capitalization stocks S&P 500 (28.6%), followed by CSI 300 (17.1%), China Market (14.3%), and TPX 100 (8.6%).
— For the regression task, **Mean Absolute Error (MAE)** and **Root-mean-squared Error (RMSE)** are the most commonly used parameters. The regression models are mostly applied to the indices of the of the Chinese stock market (50%), followed by the S&P 500 (16.7%).
— For stock recommendation tasks, MRR, IRR, and Sharpe ratio are widely used to assess the decision making capabilities of a model. The most commonly used datasets are NASDAQ (30%), NYSE (20%), CSI 100 (10%), CSI 300 (10%) and Taiwan market (10%).[9]

8 Future Direction and Challenges

In this section, based on the review done, we highlight some future research directions from the perspectives of financial knowledge graph construction, node specific features, dataset, tasks, modules, and interpretability of graph-based approaches for the stock market.

(1) Financial knowledge graph construction: Collecting relational data to build a financial knowledge graph is a difficult task due to privacy considerations and requires human efforts to study the financial reports and websites. These relational data include industry-sector, shareholding, partnership, subsidiary, or other meta-level information to establish the connections. For the stock market indices of developed economies (S&P 500, NASDAQ, NYSE, and CSI), such multi-relational data is readily available on the WikiData website. For emerging economies, collecting such data and studying their effects on prediction performance remains an open problem. The various approaches considering pre-defined multiple relationships have shown improved performance [31, 39, 60, 70] for different tasks. Although these static relationships lack the ability to reflect the real insights of the stock market. To remove the dependency on pre-defined relationships, various models have considered a data-driven approach to build a financial knowledge graph [150–152] from the market data, but it still lacks fine-grained information to study the volatile and evolving nature of the stock market.

To address this problem, meta path learning approaches hold a promising solution to identify the complex relational dependencies from the graph structured data to generate new informative graph structures [156]. It has been successfully applied in domains ranging from drug discovery to recommendation systems [89, 96]. The application of meta learning approaches in the stock market can enhance the ability of GNNs to infer implicit connections present in the financial graph constructed with available sources. This will alleviate the need to collect multi-relational data and will also reflect market situations efficiently.

[9]The percentage value in ( ) represents the proportion of the work using the respective index within the reviewed work.

(2) Node specific features: The proper selection of node features plays a crucial role in determining a model's effectiveness and accuracy in the stock market. Considering the volatile nature of stock prices, it becomes essential to encode information from multiple sources. The modeling of the stock market as a network uncovers hidden spatial dependencies, improving time-series prediction. The ability of GNNs can be further harnessed by having rich and meaningful node (stock) features that can allow graph models to distinguish between similar and dissimilar nodes, enhancing their learning capabilities. The most commonly used node features are market, fundamental, and technical data that are specific to a particular stock and reflect the economic health of a stock [92]. Choosing the perfect combination of the features poses a challenge, as a small number of features will cause information loss, and a large number of features will lead to over fitting. Several feature selection techniques have been applied to extract rich and meaningful features from a diverse set of input variables that assume independent features [58]. However, node features in financial graphs are highly correlated and redundant, so special approaches are required to select discriminate features or define node aggregation techniques to avoid instabilities in a graph model. Also in the future, the integration and optimization of information from diverse indicators such as technical, fundamental, and sentiment will be a promising solution to build a rich and meaningful node specific feature set.

The right combination of financial knowledge graph construction and node feature aggregation techniques holds immense potential for enhancing decision-making in stock market forecasting.

(3) Dataset: As discussed in Section 7.2.1, most of the graph prediction models have been applied to the dataset consisting of large cap stocks, with little or no attention to the mid and small capitalization stocks. But these stocks have the advantage of great growth potential, diversification, and affordability over large cap stocks. The market data availability of mid and small cap stocks is limited due to lower liquidity, transparency, and financial disclosures. But to achieve better performance, an abundant amount of labelled data is required to train a GNN model. In these problem settings, few shot learning offers a promising solution to analyze the data scarcity problem of mid and small cap stocks and perform node-level as well as edge-level relation prediction tasks. These few shot learning techniques have been successfully applied for applications related to the image or text domain, allowing models to generalize, adapt, and identify new concepts when data availability is limited [45, 52, 122].

The application of these approaches to non-Euclidean graph structured data poses several challenges due to its inherent complexities [154, 157]. Moreover, the recent few shot learning approaches for graphs have been studied for static, homogeneous graphs only, and their application for the stock market domain is relatively new. This requires special approaches to capture the dynamic and volatile nature of the stock data while addressing the data scarcity issue as well. The application of graph models to indices consisting of mid and small cap stocks will uncover valuable insights for investors and improve their decision making abilities for profitable investments.

(4) Tasks: Most of the work is done in the direction of node-level tasks, where the attributes or labels of nodes (stocks) are predicted. With graph neural networks, some advanced-level tasks such as graph classification [80, 160] and link prediction [159] are also possible and are being explored for different domains. By building a hybrid model that can perform node and graph classification for the financial domain, we can make predictions on the overall stock index as well, which will help economists scrutinize the condition of the overall economy. Some pooling or aggregation modelling techniques [26, 149] will be required for the graph classification task. Other than this, an interesting direction can be considered a link prediction task that can extract the hidden concept underlying the relationship between stocks that cannot be predefined. The application of this is in the stock recommendation task. In the stock market, apart from forecasting or recommendations, portfolio optimization and trading strategy simulations are other applications for graph modeling techniques. **Deep Reinforcement Learning (DRL)** offers distinct advantages when applied to trading simulation and portfolio optimization [126]. The combination of DRL and GNNs can enhance the ability of RL agents to learn complex and dynamic market situations efficiently by leveraging graph-based insights and should be explored for stock market analysis and real time applications.

(5) Modules: To enhance the prediction power, the historical information encoder and relational modules should be efficient. The historical encoder should be able to capture the long range dependencies and complex interactions between various time steps. The reviewed work in Section 7.1 suggests that the majority of the works have utilized models such as LSTM, GRU, or CNNs to encode the historical information. But the combination of these deep models with attention mechanisms has the ability to retrieve selective information and improve contextual understanding [107]. Recent advancements suggest that the self-attention-based transformer models are a suitable choice to tailor a temporal dimension due to their ability to extract complex temporal patterns and their parallel processing capabilities [138] for time series forecasting. There have been attempts to apply transformer models for stock market forecasting [94] as well. Although the complexity of models grows quadratically with the input sequence length, various solutions have been proposed to make them scalable without compromising their performance [28, 42, 166]. Combining transformers or attention-based architectures with GNNs will be an exciting approach to stock market forecasting.

Apart from that, various graph embedding models such as **Graph Auto Encoder (GAE)** [137] and its variant **Variational Graph Auto Encoder (VGAE)** [72] should be explored to learn the underlying non-linear and volatile data distribution of non-Euclidean structure. These advanced-level graph models have been successful for applications ranging from social network analysis to molecular graphs [12]. Reference [48] implies that features extracted from the VAE model achieved higher performance compared to other baseline models for the stock market. The application of GAEs in the stock market domain will help understand market instabilities and anomalies.

(6) Interpretability: GNNs are a type of deep learning model that is distinguished by its black-box structure and lack of interpretability. Different methods are available to explain deep networks for image and text data; for graph data, Explainable GNNs are experiencing rapid development [155]. Given the nature of financial data, the explainability of GNN models will be extremely useful in improving prediction performance and enhancing user trust.

9 Conclusion

Predicting the stock market is a challenging task due to the unpredictable and chaotic nature of financial time series data. In the past, various statistical, machine learning, and deep learning approaches have been proposed for this task. Deep learning models are particularly well-suited for this problem due to their ability to adapt to new data and extract relevant features automatically. However, these models have traditionally focused on capturing temporal dependencies within the data, ignoring the relational factors that can also influence stock prices. These factors can include supplier-consumer relationships, sector-industry correlations, partnerships, shareholding, and more. By treating the stock market as a network rather than a collection of isolated entities, it is possible to better understand the effects of these relational factors on stock prices. GNNs, which are a type of deep learning model, have been used successfully in a variety of other domains, such as social networks, recommendation systems, and drug discovery, and are now being applied to financial forecasting as well. GNNs are well-suited for modeling relational dependencies and have shown promise in financial forecasting tasks.

The goal of this work is to summarize recent developments in the use of graph-based techniques for stock market forecasting. We have outlined a general framework that includes various components to help researchers understand the problem. The datasets, features, and evaluation parameters used in this area are also discussed in detail. In addition, we have described the various graph models and objectives that have been used in this context and highlighted future directions and challenges for the use of graph-based approaches in stock market forecasting.

References

[1] Fatai Adebayo, R. Sivasamy, and Kehinde Shangodoyin. 2014. Forecasting stock market series with ARIMA model. *Statistical and Econometric Methods* 3 (Aug. 2014), 65 – 77.

[2] Ayodele Adebiyi, Aderemi Adewumi, and Charles Ayo. 2014. Stock price prediction using the ARIMA model. In *Proceedings of the UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (UKSim'14).*

[3] Mohammed Alshara. 2022. Multilayer graph-based deep learning approach for stock price prediction. *Secur. Commun. Netw.* 2022 (May 2022).

[4] Rodrigue Andrawos. 2022. NLP in Stock Market Prediction: A Review. Retrieved from https://doi.org/10.13140/RG.2.2.17142.68160

[5] Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. https://arxiv.org/abs/1908.10063

[6] B. Shantha and Seetha Ram Vedantam. 2019. Influence of news on rational decision making by financial market investors. *Invest. Manage. Financial Innov.* 16 (Aug. 2019), 142–156. https://doi.org/10.21511/imfi.16(3).2019.14

[7] B. Uma, D. Sundar, and P. Alli. 2013. An effective time series analysis for stock trend prediction using ARIMA model for nifty Midcap-50. *Int. J. Data Mining Knowl. Manage. Process* 3 (Jan. 2013), 65–78. https://doi.org/10.5121/ijdkp.2013.3106

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. Retrieved from https://arxiv.org/abs/1409.0473. https://doi.org/10.48550/ARXIV.1409.0473

[9] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. Retrieved from https://arxiv.org/abs/1803.01271. https://doi.org/10.48550/ARXIV.1803.01271

[10] Suzana Baresa, Sinisa Bogdan, and Zoran Ivanovic. 2013. Strategy of stock valuation by fundamental analysis. *UTMS J. Econ.* 4, 1 (2013), 45–51.

[11] Selcuk Bayraci, Yakup Ari, and Yavuz Yildirim. 2011. *A Vector Auto-Regressıve (VAR) Model for the Turkish Financial Markets.* MPRA Paper 30475. University Library of Munich, Germany. Retrieved from https://ideas.repec.org/p/pra/mprapa/30475.html

[12] Kamal Berahmand, Fatemeh Daneshfar, Elaheh Salehi, Yuefeng Li, and Yue Xu. 2024. Autoencoders and their applications in machine learning: A survey. *Artific. Intell. Rev.* 57 (Feb. 2024). https://doi.org/10.1007/s10462-023-10662-6

[13] Wolfgang Bessler and Peter Lückoff. 2008. Predicting stock returns with bayesian vector autoregressive models. In *Data Analysis, Machine Learning and Applications*, Christine Preisach, Hans Burkhardt, Lars Schmidt-Thieme, and Reinhold Decker (Eds.). Springer, Berlin, 499–506.

[14] Indronil Bhattacharjee and Pryonti Bhattacharja. 2019. Stock price prediction: A comparative study between traditional statistical approach and machine learning approach. Retrieved from https://ieeexplore.ieee.org/document/9068850. https://doi.org/10.1109/EICT48899.2019.9068850

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems* 26 (2013).

[16] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2013. Spectral networks and locally connected networks on graphs. https://arxiv.org/abs/1312.6203

[17] Chun Cai, Qinghua Ma, and Shuqiang Lv. 2012. *Research on Support Vector Regression in the Stock Market Forecasting.* Springer, Berlin, 607–612.

[18] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. Retrieved from https://arxiv.org/abs/1803.11175. https://doi.org/10.48550/ARXIV.1803.11175

[19] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2019. Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View. Retrieved from https://arxiv.org/abs/1909.03211. https://doi.org/10.48550/ARXIV.1909.03211

[20] Jun-Hao Chen and Yun-Cheng Tsai. 2022. Dynamic Deep Convolutional Candlestick Learner. https://arxiv.org/abs/2201.08669

[21] Qinkai Chen and Christian-Yann Robert. 2022. Multivariate realized volatility forecasting with graph neural network. In *Proceedings of the 3rd ACM International Conference on AI in Finance.* ACM. https://doi.org/10.1145/3533271.3561663

[22] Sheng Chen and Hongxiang He. 2018. Stock prediction using convolutional neural network. *IOP Conf. Series: Mater. Sci. Eng.* 435 (Nov. 2018), 012026. https://doi.org/10.1088/1757-899X/435/1/012026

[23] Wei Chen, Manrui Jiang, Wei-Guo Zhang, and Zhensong Chen. 2021. A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Info. Sci.* 556 (2021), 67–94.

[24] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. 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.* 1655–1658. https://doi.org/10.1145/3269206.3269269

[25] Rui Cheng and Qing Li. 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence.* 55–62.

[26] Mark Cheung, John Shi, Lavender Jiang, Oren Wright, and Jose Moura. 2019. Pooling in graph convolutional neural networks. Retrieved from https://ieeexplore.ieee.org/document/9048796. https://doi.org/10.1109/IEEECONF44664.2019.9048796

[27] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Retrieved from https://arxiv.org/abs/1406.1078. https://doi.org/10.48550/ARXIV.1406.1078

[28] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy J. Colwell, and Adrian Weller. 2020. Rethinking attention with performers. Retrieved from https://arxiv.org/abs/2009.14794

[29] Matheus José Silva de Souza, Danilo Guimarães Franco Ramos, Marina Garcia Pena, Vinicius Amorim Sobreiro, and Herbert Kimura. 2018. Examination of the profitability of technical analysis based on moving average strategies in BRICS. *Finan. Innov.* 4, 1 (2018), 1–18.

[30] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16).* Curran Associates Inc., Red Hook, NY, USA, 3844–3852.

[31] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Pan, and Huajun Chen. 2019. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. *Companion Proceedings of The 2019 World Wide Web Conference.* Association for Computing Machinery, 678–685. https://doi.org/10.1145/3308560.3317701

[32] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[33] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING'16).* The COLING Organizing Committee, 2133–2142. Retrieved from https://aclanthology.org/C16-1201

[34] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2023. Benchmarking graph neural networks. *Journal of Machine Learning Research* 24, 43 (2023), 1–48. https://doi.org/10.48550/ARXIV.2003.00982

[35] Chawwalit Faijareon and Ohm Sornil. 2019. Evolving and combining technical indicators to generate trading strategies. In *Journal of Physics: Conference Series*, Vol. 1195. IOP Publishing, 012010.

[36] Eugene F. Fama. 1970. Efficient capital markets: A review of theory and empirical work. *J. Finance* 25, 2 (1970), 383–417. Retrieved from http://www.jstor.org/stable/2325486

[37] Kamaladdin Fataliyev, Aneesh Sreevallabh Chivukula, Mukesh Prasad, and An Liu. 2021. Text-based stock market analysis: A review. https://arxiv.org/abs/2106.12985

[38] Zahra Fathali, Zahra Kodia Aouina, and Lamjed Ben Said. 2022. Stock market prediction of NIFTY 50 index applying machine learning techniques. *Appl. Artific. Intell.* 36 (Sept. 2022). https://doi.org/10.1080/08839514.2022.2111134

[39] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Trans. Info. Syst.* 37, 2 (Apr. 2019), 1–30.

[40] Shibo Feng, Chen Xu, Yu Zuo, Guo Chen, Fan Lin, and Jianbing XiaHou. 2022. Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recogn.* 121 (2022), 108119.

[41] M. C. Findlay and E. E. Williams. 2000. A fresh look at the efficient market hypothesis: How the intellectual history of finance encouraged a real "Fraud-on-the-Market." *J. Post Keynesian Econ.* 23, 2 (2000), 181–199. http://www.jstor.org/stable/4538722

[42] Quentin Fournier, Gaétan Marceau Caron, and Daniel Aloise. 2023. A Practical Survey on Faster and Lighter Transformers. *ACM Comput. Surv.* 55, Article no: 304 (2023). Association for Computing Machinery. https://doi.org/10.1145/3586074

[43] Jianliang Gao, Xiaoting Ying, Cong Xu, Jianxin Wang, Shichao Zhang, and Zhao Li. 2021. Graph-based stock recommendation by time-aware relational attention network. *ACM Trans. Knowl. Discov. Data* 16, 1, Article 4 (July 2021), 21 pages.

[44] Ya Gao, Rong Wang, and Enmin Zhou. 2021. Stock prediction based on optimized LSTM and GRU models. *Sci. Program.* 2021 (09 2021), 1–8. https://doi.org/10.1155/2021/4055281

[45] Yao Ge, Yuting Guo, Sudeshna Das, Mohammed Ali Al-Garadi, and Abeed Sarker. 2023. Few-shot learning for medical text: A review of advances, trends, and opportunities. *J. Biomed. Info.* 144, August (2023), 1–21. https://doi.org/10.1016/j.jbi.2023.104458

[46] Himanshu Goel and Narinder Singh. 2021. Dynamic prediction of Indian stock market: An artificial neural network approach. *Int. J. Ethics Syst.* 38, 1 (Aug. 2021), 35–46. https://doi.org/10.1108/IJOES-11-2020-0184

[47] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 855–864.

[48] Hakan Gunduz. 2021. An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination. *Finan. Innov.* 7, 1 (Apr. 2021), 28. https://doi.org/10.1186/s40854-021-00243-3

[49] Armin Haller, Axel Polleres, Daniil Dobriy, Nicolas Ferranti, and Sergio Rodrıguez Méndez. 2022. An analysis of links in WikiData. *European Semantic Web Conference.* Springer, 21–38.

[50] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17).* Long Beach, California, USA, Curran Associates Inc. 1025–1035. https://doi.org/10.48550/ARXIV.1706.02216

[51] Jing Hao, Feng He, Feng Ma, Shibo Zhang, and Xiaotao Zhang. 2023. Machine learning vs deep learning in stock market investment: An international evidence. *Ann. Oper. Res.* (Mar. 2023), 1–23. https://doi.org/10.1007/s10479-023-05286-6

[52] Kai He, Nan Pu, Mingrui Lao, and Michael S. Lew. 2023. Few-shot and meta-learning methods for image understanding: A survey. *Int. J. Multimedia Info. Retriev.* 12, 2 (2023), 14.

[53] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (nov 1997), 1735–1780.

[54] Ehsan Hoseinzade and Saman Haratizadeh. 2019. CNNPred: CNN-based Stock Market Prediction using Several Data Sources. *Expert Systems with Applications* 129 (2019), 273–285. https://doi.org/10.1016/j.eswa.2019.03.029

[55] Xiurui Hou, Kai Wang, Cheng Zhong, and Zhi Wei. 2021. ST-Trader: A spatial-temporal deep neural network for modeling stock market movement. *IEEE/CAA J. Autom. Sinica* 8 (May 2021), 1015–1024. https://doi.org/10.1109/JAS.2021.1003976

[56] Kanglin Hsieh, Yinyin Wang, Luyao Chen, Zhongming Zhao, Sean Savitz, Xiaoqian Jiang, Jing Tang, and Yejin Kim. 2020. Drug Repurposing for COVID-19 using Graph Neural Network with Genetic, Mechanistic, and Epidemiological Validation. *Research Square* (2020). American Journal Experts. https://doi.org/10.21203/rs.3.rs-114758/v1

[57] Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. 2023. FinGAT: Financial Graph Attention Networks for Recommending Top-K Profitable Stocks. *IEEE Transactions on Knowledge and Data Engineering* 35 (2023), 469–481. DOI: 10.1109/TKDE.2021.3079496

[58] Htet Htun, Michael Biehl, and Nicolai Petkov. 2023. Survey of feature selection and extraction techniques for stock market prediction. *Finan. Innov.* 9 (Jan. 2023), 26. https://doi.org/10.1186/s40854-022-00441-7

[59] Zexin Hu, Yiqi Zhao, and Matloob Khushi. 2021. A survey of forex and stock price prediction using deep learning. *Appl. Syst. Innov.* 4, 1 (Feb. 2021), 9. https://doi.org/10.3390/asi4010009

[60] Kun Huang, Xiaoming Li, Fangyuan Liu, Xiaoping Yang, and Wei Yu. 2022. ML-GAT: A multilevel graph attention model for stock prediction. *IEEE Access* 10 (2022), 86408–86422.

[61] Wenbing Huang, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. 2022. Tackling Over-Smoothing for General Graph Convolutional Networks. https://arxiv.org/abs/2008.09864

[62] Pavan Kumar Illa, Balakesavareddy Parvathala, and Anand Kumar Sharma. 2022. Stock price prediction methodology using random forest algorithm and support vector machine. *Mater. Today: Proc.* 56 (2022), 1776–1782.

[63] Alireza Jafari and Saman Haratizadeh. 2022. GCNET: Graph-based prediction of stock price movement using graph convolutional network. *Eng. Appl. Artific. Intell.* 116 (2022), 105452. https://doi.org/10.1016/j.engappai.2022.105452

[64] Xuan Ji, Jiachen Wang, and Zhijun Yan. 2021. A stock price prediction method based on deep learning technology. *Int. J. Crowd Sci.* (Mar. 2021). https://doi.org/10.1108/IJCS-05-2020-0012

[65] Weiwei Jiang. 2021. Applications of deep learning in stock market prediction: Recent progress. *Expert Syst. Appl.* 184 (Dec. 2021), 115537. https://doi.org/10.1016/j.eswa.2021.115537

[66] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* 207 (2022), 117921. https://doi.org/10.1016/j.eswa.2022.117921

[67] Peiran Jiao, André Veiga, and Ansgar Walther. 2020. Social media, news media and the stock market. *J. Econ. Behav. Org.* 176 (2020), 63–90. https://doi.org/10.1016/j.jebo.2020.03.002

[68] Ye Jiexia, Juanjuan Zhao, Kejiang Ye, and Cheng-Zhong Xu. 2021. Multi-View Graph Convolutional Networks for Relationship-Driven Stock Prediction. *2020 25th International Conference on Pattern Recognition (ICPR).* IEEE, 6702–6709. http://dx.doi.org/10.1109/ICPR48806.2021.9412695

[69] Dror Y. Kenett, Xuqing Huang, Irena Vodenska, Shlomo Havlin, and H. Eugene Stanley. 2015. Partial Correlation Analysis: Applications for Financial Markets. *Quantitative Finance* 15, 4 (2015), 569–578. https://doi.org/10.48550/ARXIV.1402.1405

[70] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. 2019. HATS: A Hierarchical Graph Attention Network for Stock Movement Prediction. https://arxiv.org/abs/1908.07999

[71] Thomas Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. (09 2016). https://arxiv.org/abs/1609.02907

[72] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. https://arxiv.org/abs/1611.07308

[73] Ching-Ru Ko and Hsien-Tsung Chang. 2021. LSTM-based sentiment analysis for stock price forecast. *PeerJ Comput. Sci.* 7 (Mar. 2021), e408.

[74] Paraskevas Koukaras, Christina Nousi, and Christos Tjortjis. 2022. Stock market prediction using microblogging sentiment analysis and machine learning. *Telecom* 3, 2 (2022), 358–378.

[75] M. Kumar and N. V. Manoj. 2020. Market capitalization: Pre and post COVID-19 analysis. *Mater. Today: Proc.* 37 (Oct. 2020). https://doi.org/10.1016/j.matpr.2020.08.493

[76] Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, and Jari Porras. 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Syst. Appl.* 197 (2022), 116659. https://doi.org/10.1016/j.eswa.2022.116659

[77] Rosdyana Mangir Irawan Kusuma, Trang-Thi Ho, Wei-Chun Kao, Yu-Yen Ou, and Kai-Lung Hua. 2019. Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market. https://arxiv.org/abs/1903.12258

[78] Patrick Laub, Thomas Taimre, and Philip Pollett. 2015. Hawkes processes. https://arxiv.org/abs/1507.02822

[79] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. *International Conference on Machine Learning*. PMLR, 1188–1196. https://doi.org/10.48550/ARXIV.1405.4053

[80] John Lee, Ryan Rossi, and Xiangnan Kong. 2018. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1666–1674. https://doi.org/10.1145/3219819.3219980

[81] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2020. Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'20)*, Christian Bessiere (Ed.). 4541–4547.

[82] Xiaohan Li, Jun Wang, Jinghua Tan, Shiyu Ji, and Huading Jia. 2022. A graph neural network-based stock forecasting method utilizing multi-source heterogeneous data fusion. *Multimedia Tools and Applications* 81, 30 (2022), 43753–43775.

[83] K. Liagkouras and K. Metaxiotis. 2020. *Stock Market Forecasting by Using Support Vector Machines*. Springer International Publishing, Cham, 259–271.

[84] Fabrizio Lillo, Salvatore Micciché, Michele Tumminello, Jyrki Piilo, and Rosario N. Mantegna. 2015. How news affects the trading behaviour of different categories of investors in a financial market. *Quant. Finance* 15, 2 (2015), 213–229. https://doi.org/10.1080/14697688.2014.931593 arXiv:https://doi.org/10.1080/14697688.2014.931593

[85] Ching Lim and Sek Siok Kun. 2013. Comparing the performances of GARCH-type models in capturing the stock market volatility in Malaysia. *Procedia Econ. Finance* 5 (Dec. 2013), 478–487. https://doi.org/10.1016/S2212-5671(13)00056-7

[86] Yang Lin, Irena Koprinska, and Mashud Rana. 2021. Temporal convolutional attention neural networks for time series forecasting. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'21)*. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534351

[87] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*. AAAI Press, 2181–2187.

[88] Jue Liu, Zhuocheng Lu, and W. P. Du. 2019. Combining enterprise knowledge graph and news sentiment analysis for stock price prediction. In *Proceedings of the Hawaii International Conference on System Sciences*.

[89] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Learning to Propagate for Graph Meta-Learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* Article No.: 94 (2019), 1039–1050.

[90] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. 2020. A CNN-LSTM-based model to forecast stock prices. *Complexity* 2020 (Nov. 2020), 6622927.

[91] Dongbo Ma, Da Yuan, Maojun Huang, and Ling Dong. 2024. VGC-GAN: A multi-graph convolution adversarial network for stock price prediction. *Expert Syst. Appl.* 236 (2024), 121204.

[92] Mahinda Mailagaha Kumbure, Christoph Lohrmann, Pasi Luukka, and Jari Porras. 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Syst. Appl.* 197 (02 2022), 116659.

[93] Spyros Makridakis, Evangelos Spiliotis, Vassilis Assimakopoulos, Artemios-Anargyros Semenoglou, Gary Mulder, and Konstantinos Nikolopoulos. 2022. Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward. *J. Oper. Res. Soc.* 74 (Sept. 2022), 1–20. https://doi.org/10.1080/01605682.2022.2118629
[94] Nadeem Malibari, Iyad Katib, and Rashid Mehmood. 2021. Predicting stock closing prices in emerging markets with transformer neural networks: The Saudi stock exchange case. *Int. J. Adv. Comput. Sci. Appl.* 12 (Jan. 2021).
[95] Burton G. Malkiel. 2003. The efficient market hypothesis and its critics. *J. Econ. Perspect.* 17, 1 (Mar. 2003), 59–82.
[96] Debmalya Mandal, Sourav Medya, Brian Uzzi, and Charu Aggarwal. 2022. MetaLearning with graph neural networks: Methods and applications. *SIGKDD Explor. Newsl.* 23, 2 (Jan. 2022), 13–22. https://doi.org/10.1145/3510374.3510379
[97] Arfa Maqsood, Suboohi Safdar, Rafia Shafi, and Ntato Lelit. 2017. Modeling stock market volatility using GARCH models: A case study of Nairobi Securities Exchange (NSE). *Open J. Stat.* 07 (Jan. 2017), 369–381. https://doi.org/10.4236/ojs.2017.72026
[98] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. 2019. Exploring graph neural networks for stock market predictions with rolling window analysis. https://arxiv.org/abs/1909.10660
[99] Marcelo C. R. Melo, Jacqueline R. M. A. Maasch, and Cesar de la Fuente-Nunez. 2021. Accelerating antibiotic discovery through artificial intelligence. *Commun. Biol.* 4, 1 (09 Sep 2021), 1050.
[100] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013*, Workshop Track Proceedings. http://arxiv.org/abs/1301.3781
[101] Adil Moghar and Mhamed Hamiche. 2020. Stock market prediction using LSTM recurrent neural network. *Procedia Comput. Sci.* 170 (2020), 1168–1173.
[102] C. Mohankumari, M. Vishukumar, and N. R. Chillale. 2019. Analysis of daily stock trend prediction using ARIMA model. *International Journal of Mechanical Engineering and Technology* 10 (Jan. 2019), 1772–1792.
[103] Oweys Momenzada, Michael Palk, and Stefan Voss. 2021. Graph Neural Networks for Efficient Recommender Systems. https://www.researchgate.net/publication/355928151_Graph_Neural_Networks_for_Efficient_Recommender_Systems
[104] Marisetty Nagendra and Dr Haritha. 2014. NSE NIFTY and its correlation with sectorial indexes. *International Journal of Conceptions on Management and Social Sciences* 2, 1 (2014).
[105] Nagaraj Naik, Biju Mohan, and Rajat Jha. 2020. GARCH model identification for stock crises events. *Procedia Comput. Sci.* 171 (Jan. 2020), 1742–1749. https://doi.org/10.1016/j.procs.2020.04.187
[106] Sobia Naseem, Muhammad Mohsin, Wang Hui, Geng Liyan, and Kun Penglai. 2021. The investor psychology and stock market behavior during the initial era of COVID-19: A study of China, Japan, and the United States. *Front. Psychol.* 12 (Feb. 2021).
[107] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. 2021. A review on the attention mechanism of deep learning. *Neurocomputing* 452 (Apr. 2021). https://doi.org/10.1016/j.neucom.2021.03.091
[108] Abdullah Bin Omar, Shuai Huang, Anas A. Salameh, Haris Khurram, and Muhammad Fareed. 2022. Stock market forecasting using the random forest and deep neural network models before and during the COVID-19 period. *Front. Environ. Sci.* 10 (2022). https://doi.org/10.3389/fenvs.2022.917047
[109] Felipe Oriani and Guilherme Coelho. 2016. Evaluating the impact of technical indicators on stock forecasting. *2016 IEEE Symposium Series on Computational Intelligence (SSCI).* IEEE, 1–8. https://doi.org/10.1109/SSCI.2016.7850017
[110] Mihály Ormos and Miklós Vázsonyi. 2011. Impacts of public news on stock market prices: Evidence from S&P5001. *Interdisc. J. Res. Bus.* 1 (Mar. 2011), 1–17.
[111] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *EMNLP* 14, 1532–1543.
[112] Pramod and Mallikarjuna Pm. 2021. Stock price prediction using LSTM. *Test Eng. Manage.* 83 (Jan. 2021), 5246–5251.
[113] R. D. Vasudevan and S. Vetrivel. 2016. Forecasting stock market volatility using GARCH models: Evidence from the Indian stock market. *Asian J. Res. Soc. Sci. Human.* 6 (Jan. 2016), 1565.
[114] João Rico, José Barateiro, and Arlindo Oliveira. 2021. Graph Neural Networks for Traffic Forecasting. https://arxiv.org/abs/2104.13096
[115] Nusrat Rouf, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheol Kim. 2021. Stock market prediction using machine learning techniques: A decade survey on methodologies, recent developments, and future directions. *Electronics* 10, 21 (2021). Retrieved from https://www.mdpi.com/2079-9292/10/21/2717
[116] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 1 (2021), 497–504. https://doi.org/10.1609/aaai.v35i1.16127
[117] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. 2021. Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proceedings of the Web Conference (WWW '21).* Association for Computing Machinery, New York, NY, 11–22. https://doi.org/10.1145/3442381.3450095

[118] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn Shah. 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 8415–8426. https://doi.org/10.18653/v1/2020.emnlp-main.676

[119] Dharmaraja Selvamuthu, Vineet Kumar, and Abhishek Mishra. 2019. Indian stock market prediction using artificial neural networks on tick data. *Finan. Innov.* 5, 1 (2019), 1–12. https://doi.org/10.1186/s40854-019-0131-7

[120] Atharva Shah, Maharshi Gor, Meet Sagar, and Manan Shah. 2022. A stock market trading framework based on deep learning architectures. *Multimedia Tools Appl.* 81 (Apr. 2022), 1–19. https://doi.org/10.1007/s11042-022-12328-x

[121] Dev Shah, Haruna Isah, and Farhana Zulkernine. 2019. Stock market analysis: A review and taxonomy of prediction techniques. *Int. J. Finan. Studies* 7, 2 (2019). Retrieved from https://www.mdpi.com/2227-7072/7/2/26

[122] Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys* 55, 13s, Article 271 (July 2023), 40 pages. https://doi.org/10.1145/3582688

[123] Priyank Sonkiya, Vikas Bajpai, and Anukriti Bansal. 2021. Stock Price Prediction using BERT and GAN. https://arxiv.org/abs/2107.09055

[124] Troy J. Strader, John J. Rozycki, Thomas H. Root, and Yu-Hsiang Huang. 2020. Machine learning stock market prediction studies: Review and research directions. *J. Int. Technol. Info. Manage.* 28 (2020), 63–83.

[125] Jifeng Sun, Jianwu Lin, and Yi Zhou. 2020. Multi-channel temporal graph convolutional network for stock return prediction. *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)* 1 (2020). 423–428. https://doi.org/10.1109/INDIN45582.2020.9442196

[126] Shuo Sun, Rundong Wang, and Bo An. 2023. Reinforcement learning for quantitative trading. *ACM Transactions on Intelligent Systems and Technology* 14, 3 (2023).

[127] Meiyao Tao, Shanshan Gao, Deqian Mao, and Hong Huang. 2022. Knowledge graph and deep learning combined with a stock price prediction network focusing on related stocks and mutation points. *J. King Saud Univ. Comput. Info. Sci.* 34, 7 (2022), 4322–4334.

[128] Amit Tewari. 2020. Forecasting NIFTY 50 Benchmark Index using Seasonal ARIMA Time Series Models. DOI: 10.13140/RG.2.2.10332.95364

[129] Hu Tian, Xingwei Zhang, Xiaolong Zheng, and Daniel Dajun Zeng. 2023. Learning dynamic dependencies with graph evolution recurrent unit for stock predictions. *IEEE Trans. Syst. Man Cybernet.: Syst.* 53, 11 (2023), 6705–6717. https://doi.org/10.1109/TSMC.2023.3284840

[130] Yassine Touzani and Khadija Douzi. 2021. An LSTM and GRU based trading strategy adapted to the Moroccan market. *J. Big Data* 8, 1 (24 Sep 2021), 126.

[131] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. https://arxiv.org/abs/1710.10903

[132] Denny Vrandecić and Markus Krötzsch. 2014. WikiData: A free collaborative knowledgebase. *Commun. ACM* 57, 10 (Oct. 2014), 2014.

[133] Ahmed Wafi, Hassan Hassan, and Adel Mabrouk. 2015. Fundamental analysis models in financial markets – review study. *Procedia Econ. Finance* 30 (Nov. 2015), 939–947. https://doi.org/10.1016/S2212-5671(15)01344-1

[134] Changhai Wang, Hui Liang, Bo Wang, Xiaoxu Cui, and Yuwei Xu. 2022. MG-Conv: A spatiotemporal multi-graph convolutional neural network for stock market index trend prediction. *Comput. Electr. Eng.* 103 (2022), 108285.

[135] Heyuan Wang, Shun Li, Tengjiao Wang, and Jiayi Zheng. 2021. Hierarchical adaptive temporal-relational modeling for stock trend prediction. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence IJCAI-21*, Zhi-Hua Zhou (Ed.). 3691–3698. https://doi.org/10.24963/ijcai.2021/508

[136] H. Wang, T. Wang, S. Li, and S. Guan. 5555. HATR-I: Hierarchical adaptive temporal relational interaction for stock trend prediction. *IEEE Trans. Knowl. Data Eng.* 01 (July 5555), 1–14. https://doi.org/10.1109/TKDE.2022.3188320

[137] Yingfeng Wang, Biyun Xu, Myungjae Kwak, and Xiaoqin Zeng. 2020. A simple training strategy for graph autoencoder. *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*. 341–345. https://doi.org/10.1145/3383972.3383985

[138] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. 2023. Transformers in Time Series: A Survey. Retrieved from https://arXiv:2202.07125.

[139] Yulian Wen, Peiguang Lin, and Xiushan Nie. 2020. Research of stock price prediction based on PCA-LSTM model. *IOP Conf. Ser.: Mater. Sci. Eng.* 790 (Apr. 2020), 012109. https://doi.org/10.1088/1757-899X/790/1/012109

[140] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: A survey. *Comput. Surveys* (May 2022). https://doi.org/10.1145/3535101

[141] Yangru Wu and Xing Zhou. 2015. *VAR Models: Estimation, Inferences, and Applications*. Springer, New York, NY, 2077–2091. https://doi.org/10.1007/978-1-4614-7750-1_76

[142] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip Yu. 2019. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2019), 4–24.
[143] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. 2021. Graph learning: A survey. *IEEE Trans. Artific. Intell.* 2, 2 (2021), 109–127.
[144] Huize Xu, Yuhang Zhang, and Yaoqun Xu. 2023. Promoting financial market development-financial stock classification using graph convolutional neural networks. *IEEE Access* 11 (2023), 49289–49299. https://doi.org/10.1109/ACCESS.2023.3275085
[145] Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2022. HIST: A Graph-based Framework for Stock Trend Forecasting via Mining Concept-Oriented Shared Information. https://arxiv.org/abs/2110.13716
[146] Wentao Xu, Weiqing Liu, Chang Xu, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. REST: Relational event-driven stock trend forecasting. In *Proceedings of the Web Conference.* ACM.
[147] Yan Yan, Boyao Wu, Tianhai Tian, and Hu Zhang. 2020. Development of stock networks using part mutual information and australian stock market data. *Entropy* 22, 7 (2020).
[148] Bing Yang, Jian Hao, and Si Zhang. 2012. Stock market prediction using artificial neural networks. *Adv. Eng. Forum* 6-7 (Sept. 2012), 1055–1060.
[149] Jia-Qi Yang, De-Chuan Zhan, and Xin-Chun Li. 2020. Bottom-up and top-down graph pooling. In *Advances in Knowledge Discovery and Data Mining*, Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan (Eds.). Springer International Publishing, Cham, 568–579.
[150] He Yanshen, Li Qiutong, Wu Feng, and Gao Jianliang. 2022. Static-dynamic graph neural network for stock recommendation. *Proceedings of the 34th International Conference on Scientific and Statistical Database Management.* 1–4. https://doi.org/10.1145/3538712.3538733
[151] Jain Yash, Veer Atharva, Sawant Gaurav, Jain Yash, and Naik Sowmiyaraksha. 2021. Stock prices prediction of bombay stock exchange using graph convolutional networks. *International Research Journal of Engineering and Technology (IRJET)* 8, 10 (2021).
[152] Tao Yin, Chenzhengyi Liu, Fangyu Ding, Ziming Feng, Bo Yuan, and Ning Zhang. 2022. Graph-based stock correlation and prediction for high-frequency trading systems. *Pattern Recogn.* 122 (2022), 108209. https://doi.org/10.1016/j.patcog.2021.108209
[153] Xingkun Yin, Da Yan, Abdullateef Almudaifer, Sibo Yan, and Yang Zhou. 2021. Forecasting stock prices using stock correlation graph: A graph convolutional network approach. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'21).* 1–8.
[154] Xingtong Yu, Yuan Fang, Zemin Liu, Yuxia Wu, Zhihao Wen, Jianyuan Bo, Xinming Zhang, and Steven C. H. Hoi. 2024. Few-Shot Learning on Graphs: From Meta-learning to Pre-training and Prompting. Retrieved from https://arXiv:2402.01440
[155] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 5 (2022), 5782–5799. https://doi.org/10.1109/TPAMI.2022.3204236
[156] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer Networks. Retrieved from https://arxiv.org/abs/1911.06455. https://doi.org/10.48550/ARXIV.1911.06455
[157] Chuxu Zhang, Kaize Ding, Jundong Li, Xiangliang Zhang, Yanfang Ye, Nitesh V. Chawla, and Huan Liu. 2022. Few-Shot Learning on Graphs. Retrieved from https://arxiv.org/abs/2203.09308. https://doi.org/10.48550/ARXIV.2203.09308
[158] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. 2018. GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. Retrieved from https://arxiv.org/abs/1803.07294. https://doi.org/10.48550/ARXIV.1803.07294
[159] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. Retrieved from https://arxiv.org/abs/1802.09691. https://doi.org/10.48550/ARXIV.1802.09691
[160] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v32i1.11782
[161] Qiuyue Zhang, Zhang Yunfeng, Fangxun Bao, Caiming Zhang, and Peide Liu. 2023. SMPDF: Stock movement prediction based on stock prices and text. *Int. J. Gen. Syst.* (Dec. 2023), 1–18. https://doi.org/10.1080/03081079.2023.2294132
[162] Qiuyue Zhang, Yunfeng Zhang, Xunxiang Yao, Shilong Li, Caiming Zhang, and Peide Liu. 2023. A dynamic attributes-driven graph attention network modeling on behavioral finance for stock prediction. *ACM Trans. Knowl. Discov. Data* 18, 1, Article 16 (Sep. 2023), 29 pages. https://doi.org/10.1145/3611311
[163] Xi Zhang, Yunjia Zhang, Senzhang Wang, Yuntao Yao, Binxing Fang, and Philip S. Yu. 2018. Improving stock market prediction via heterogeneous information fusion. *Knowl.-Based Syst.* 143 (2018), 236–247. https://doi.org/10.1016/j.knosys.2017.12.025

[164] Yu Zhao, Huaming Du, Ying Liu, Shaopeng Wei, Xingyan Chen, Huali Feng, Qinghong Shuai, Qing Li, Fuzhen Zhuang, and Gang Kou. 2022. Stock Movement Prediction Based on Bi-typed and Hybrid-relational Market Knowledge Graph via Dual Attention Networks. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2022), 8559–8571. DOI: 10.1109/TKDE.2022.3220520

[165] Zetao Zheng, Jie Shao, Jia Zhu, and Heng Tao Shen. 2023. Relational temporal graph convolutional networks for ranking-based stock prediction. In *Proceedings of the IEEE 39th International Conference on Data Engineering (ICDE'23)*. 123–136. https://doi.org/10.1109/ICDE55515.2023.00017

[166] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2020. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Retrieved from https://arxiv.org/abs/2012.07436. https://doi.org/10.48550/ARXIV.2012.07436

Received 25 January 2023; revised 11 April 2024; accepted 12 September 2024