# A Dynamic Dual-Graph Neural Network for Stock Price Movement Prediction

Kelvin Du[♠], Rui Mao[♠], Frank Xing[♣] and Erik Cambria[♠]
[♠]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[♣]*Department of Information Systems and Analytics, National University of Singapore, Singapore*
zidong001@e.ntu.edu.sg; rui.mao@ntu.edu.sg; xing@nus.edu.sg; cambria@ntu.edu.sg

**Abstract**

The prediction of stock price movements is challenging due to the inherently dynamic and complex characteristics of financial markets. A current research gap is the lack of exploration into the complex interrelationships inherent in stock price dynamics, often analyzing predictions in isolation with an implicit presumption that solely the historical data of a given stock influences its future trend. However, stock prices are impacted by a diverse array of driving factors that extend beyond the traditionally examined historical prices, encompassing influences such as inter-stock correlations. In this paper, we present a predictive approach using a dynamic dual-graph neural network. The network combines textual data and quantitative metrics to capture multiple dynamic relationships. Specifically, We have developed a price relationship graph (PRG) and a semantic relationship graph (SRG), which are later integrated using a graph attention neural network. The effectiveness of our neural architecture is validated through extensive testing on two benchmark datasets for stock movement prediction, illustrating its superior performance compared to other graph-based networks for stock market prediction.

**Index Terms**

Graph attention networks, financial forecasting, neural networks, natural language processing

## I. Introduction

Forecasting stock price movements is recognized as a complex endeavor, influenced by the ever-evolving and intricate dynamics of the financial market. Predominantly, stock market analysis leverages two primary approaches: fundamental and technical analysis [1]. Fundamental analysis methodically investigates the intrinsic value of stocks, encapsulating an extensive analysis of macroeconomic factors, industry conditions, and company-specific financial health, including revenue, expenses, assets, and liabilities. In contrast, technical analysis focuses on identifying trends through statistical analysis of stock prices, employing indicators like Simple Moving Average (SMA), Exponential Moving Average (EMA), and Moving Average Convergence/Divergence (MACD). Most prior research in this field has relied heavily on historical data, technical indicators, and macroeconomic factors. However, an emerging area of study integrates financial textual data with these analyses, applying NLP techniques to enhance financial forecasting [2]–[8].

A significant research gap lies in the limited exploration of the inherent relationships within stock price dynamics. The prevailing underlying assumption in the majority of previous work is that stocks function autonomously, independent of each other's influence [9]–[16], which frequently overlooked the cross-effect of stocks over time, a dynamic factor influencing stock prices. The interplay between stocks can be traced back to an array of inter-corporate connections, encompassing aspects like shared industry information, supply chain dynamics, payment networks, business partnerships, and shareholder ownership structures [17]. These intricate and diverse linkages culminate in a scenario where the price of an individual stock is not solely dependent on its own information but is also significantly correlated with the performance of other stocks.

While recent research tried to explore the use of stock relationships to predict stock price movements [5] and optimize portfolios [7], these studies have primarily concentrated on specific relation types defined by explicit corporate relationships, e.g., supply chain, shareholding chain, and industry competition relationships, which can be identified by public information. Although the identified corporate relations offer some insight into the mutual influence of related companies on their stock prices, it is crucial to acknowledge that the interaction between stock prices extends far beyond the scope of corporate relationships. For example, trading in stock indices has the potential to influence the prices of all stocks encompassed by the respective index. Additionally, when big portfolio holders engage in trading activities, it can lead to price fluctuations simultaneously in individual stocks within the portfolio. The repercussions of these dynamics extend beyond explicit corporate relationships, contributing to dynamic interactions and varying influences on stock prices across different periods. Effectively identifying and leveraging dynamic correlations, including price and information correlation, for price prediction remains an unexplored challenge in the current literature. To bridge these gaps, this study introduces a novel network architecture for predicting stock price movements, which uniquely incorporates dynamic corporate relationships in their stock prices and textual information. Our methodology synergizes both quantitative and textual data, effectively mapping out daily price and semantic relationships derived from past 7-day historical data. The identified price and semantic relationships are subsequently integrated into the model training process by adopting graph attention networks.

This technique enhances the understanding of the key drivers behind stock market movements, thereby augmenting the predictive precision for subsequent trading days. An extensive validation process was undertaken using two benchmark datasets [18], [19] to evaluate the efficacy of our model. The results indicate that our model consistently surpasses the most robust existing baseline, achieving an average accuracy enhancement of 4.5%. The conducted ablation study provides additional affirmation regarding the efficacy of our suggested approaches for price and semantic relationship generation in modeling the correlation between companies based on their price and textual information. This results in an average accuracy improvement of 3.6%.

Our contributions can be summarized into two points:

1) A novel relationship-driven network architecture is proposed, grounded in significant financial intuition, which revolves around predicting stock price movements by generating dynamic price and semantic relationships using textual and quantitative data.
2) The proposed model outperforms strong baselines on public benchmark datasets, underscoring its effectiveness in stock price movement prediction.

## II. Related Work

Research in the field of stock market prediction encompasses various aspects, such as market index, stock price, stock price movement, return rate, and volatility etc. To achieve prediction goals, time series models, machine learning techniques, deep learning approaches, and reinforcement learning methods have been explored. Specifically, Ding et al. [20] proposed a novel neural tensor network combined with a deep convolutional neural network (CNN) to predict event-driven stock price movements in the S&P 500 index and individual stocks. The accuracy and Matthews Correlation Coefficient (MCC) were employed as evaluation metrics, and the simulation results demonstrated the effectiveness of deep learning in event-driven market prediction. Zhou et al. [21] proposed event-driven trading strategies that detect corporate events, considered as driving forces of market movements, from news articles. They trained a bi-level event detection model using the masked-language model (MLM) loss. Two trading strategies were tested on the EDT dataset, with Trade at End Strategy yielding a return outperforming sentiment-based models. The Trade at Best Strategy, which completed transactions within a specified time frame, resulted in a return also surpassing all sentiment-based models. Another study by Liu et al. [22] focused on extracting features from news titles through a CNN and event tuples (comprising Agent, Predicate, and Object) via knowledge graph embedding (TransE model). These features were combined with daily trading and technical analysis data, and SVM and LSTM models were utilized for stock price movement prediction. Joint learning of event tuples and text was found to be the most effective approach, addressing the text sparsity problem in feature extraction. A deep generative model called StockNet was proposed by Xu and Cohen [23] for stock market prediction based on binary movement, denoting a rise in stock price as one and a fall as zero. This model consisted of three components: Market Information Encoder, Variational Movement Decoder, and Attentive Temporal Auxiliary. Evaluation using Accuracy and MCC demonstrated that StockNet achieved state-of-the-art performance on a new stock movement prediction dataset, which was made publicly available. Feng et al. [13] proposed to implement adversarial training as a way to enhance the predictive model's generalization capacity and achieved a significant performance improvement as compare to [23].

The Efficient Market Hypothesis posits that financial markets exhibit informational efficiency [24]. This assumption leads to the logical inference that the fluctuation in the stock prices of a specific entity can be influenced by its related corporations. Thus, previous studies leveraged explicit company relations to improve financial prediction tasks [5], [7], [25], [26]. Being a pioneer study, Chen et al. [25] incorporated top-10 stock-holding relationships of list companies, using Graph Convolutional Neural Networks (GCNN). Later, Sawhney et al. [26] proposed a deep attentive learning approach for predicting stock movements based on information from social media texts and company correlations, where company relationships were obtained from Wikidata[1]. In recent studies, Ma et al. [5], [7] also leveraged company relationships to predict stock price movements and portfolio optimization. They developed knowledge graphs based on three types of corporate relations, e.g., supply chain, shareholding chain, and industry competition relationship. An attentive GCN was proposed [7] to learn the impact of different relations. The hypothesis is that different corporate relationships may have different influences on the stock prices, e.g., the stock prices of companies with the supply chain relationship may be positively related, while stock prices may have a negative correlation between competitive companies.

However, simply modeling explicit corporate relationships may be sub-optimal in real-world investment. In the evolving landscape of the financial market, companies are extensively intertwined through diverse interconnections. In addition to the corporate relationships explored in prior research, there exist unexplored corporate associations, including strategic partnerships, joint ventures, mergers and acquisitions, collaborative research and development, license agreements, and other nuanced affiliations. The challenge lies in acquiring a comprehensive dataset of all potential relationships for timely incorporation into stock price modeling. Moreover, implicit correlations in stock prices may not necessarily manifest in explicit corporate relationships. Correlation in stock prices can occur when two prices are influenced by factors such as

[1] https://www.wikidata.org/wiki/Wikidata:List_of_properties

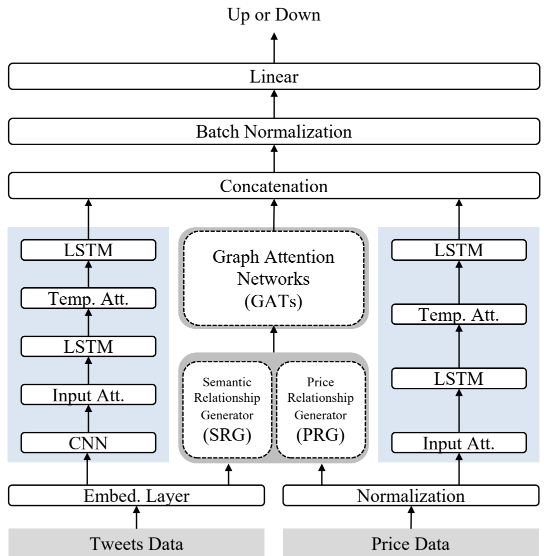


Fig. 1. Proposed Network Architecture for Dynamic Dual-Graph Neural Network for Stock Price Movement Prediction.

being under the same stock trading index or managed by the same fund manager, irrespective of direct corporate connections between the involved companies.

## III. Methodology

Contrary to the previous works, our perspective posits that relationships can be more accurately represented through dynamic means, specifically via price and semantic relationships that evolve over time. This dynamic perspective offers a novel approach to understanding and analyzing the relationships between stock prices, distinguishing our method from traditional static data analysis methods. Our research seeks to construct a dynamic mechanism that delineates the stock relationships, drawing on data from text and prices.

The proposed network architecture (Fig. 1) is composed of seven primary modules: Tweet Embedding Layer (TEL), Tweets Encoder (TE), Price Normalization Module (PNM), Price Encoder (PE), Price Relationship Generator (PRG), Semantic Relationship Generator (SRG), and Graph Attention Network (GAT). The TEL is tasked for vector representation of tweets. TE is for encoding tweet vectors using CNN and Attentive LSTM architecture. The PNM is designed for normalizing the stock price data. PE is for encoding normalized price data using Attentive LSTM architecture. The PRG and SRG modules are dedicated for constructing the dynamic price relationship graph and semantic relationship graph, respectively. Finally, the GAT module is responsible for the effective encoding of company relations, integrating the graph information produced by PRG and SRG.

In the remainder of this section, we first introduce the task definition (Section III-A), and then elaborate on the functions of each network component (Section III-B-Section III-H).

### A. Task Definition

The objective of the proposed architecture is to leverage pertinent data from both tweets and historical stock prices to forecast the stock price movements. Given the task formulation from a related work [23], this study defines stock movement based on the variation in the adjusted closing prices of a given stock between consecutive trading days (d and d + 1), which frames the prediction of stock movement as a binary classification task.

Following the methodologies in stock movement prediction [19], [23], the task is defined as follows: for a given stock $A$, considering its historical price data and related tweets across a retrospective window of $l$ days, i.e., within the range $[t-l, t]$, the future price movement of stock A from day $t$ to day $t+1$ is defined by a label of zero if the adjusted closing price movement from day $t$ to day $t+1$ is equal to or below -0.5%, and one if the adjusted closing price movement is above 0.55%. In this binary classification setting, the value $0$ indicates a price decline, while $1$ signifies an increase.

### B. Tweet Embedding Layer

Textual information can significantly impact market cognition [27] and sentiment [28]–[31], subsequently, reflected in the stock prices. We leverage tweets, drawing from datasets previously compiled in established literature [18], [19], [23], as a source of textual information to enhance the accuracy of stock price movement prediction.

A Tweet Embedding Layer is designed to specifically encode tweets and generate their corresponding embeddings. In this module, each tweet is transformed into an embedding vector $\mathbf{e} \in \mathbb{R}^n$, which is accomplished using the Sentence-BERT framework [32]. Sentence-BERT is a modification of the pre-trained BERT network which use siamese and triplet network structures, enabling it to effectively capture the semantic significance of sentence embeddings. The dimensionality of these embeddings is set at $n = 384$. Considering a scenario with $m$ tweets on a given day $i$, related to a particular stock, say stock $A$, this collection of tweets can be represented as a sequence and the TEL processes each tweet in this sequence to produce a series of vectors $[\mathbf{e}_1^i, \mathbf{e}_2^i, \ldots, \mathbf{e}_m^i]$, where $\forall \mathbf{e} \in \mathbb{R}^n$, and $m$ signifies the total count of tweets concerning stock $A$ on that day. The output generated by TEL is methodically fed as input to two subsequent components: TE and SRG.

### C. Semantic Relationship Generator

The semantic relationship is measured by the cosine similarity applied to the sentence embeddings derived from tweet data. Given company $A$ and $B$, and their corresponding tweets vectors $\{\mathbf{e}_j^i(A)\}$ and $\{\mathbf{e}_j^i(B)\}$, where $i = 1, 2, ...l$ are the day index and $j$ the intraday tweet index, the semantic relationship is defined as:

$$r_s(A, B) = \overline{\cos[\mathbf{e}_a \in \{\mathbf{e}_j^i(A)\}, \mathbf{e}_b \in \{\mathbf{e}_j^i(B)\}]}, \tag{1}$$

where we calculate pairwise cosine similarities for all tweets emanating from two companies over a preceding period of $l$ days. The mean cosine similarity is then served as a quantitative representation of the relational proximity between the tweet collections of the respective companies.

### D. Tweets Encoder

The TE module mainly consists of a CNN paired with a dual-stage Attentive Long Short-Term Memory (ALSTM) network, strategically engineered to effectively capture and compress contextual and sequential dependencies.

The TEL module encodes tweets over $l$ days into $\mathbf{E} \in \mathbb{R}^{l \times m \times n}$, where $m$ denotes the maximum number of daily tweets; $n$ represents the dimensionality of the sentence embedding. $\mathbf{E}$ serves as input to a 2D-CNN

$$\mathbf{C} = \text{Conv2d}(\mathbf{E}) \tag{2}$$

Each convolution involves using a filter $w$ from the space $\mathbb{R}^{k \times 1}$, with $k$ taking the values $3, 4, 5$. The number of input and output channels are uniformly set to $l$. A unique feature $c_{ij} \in \mathbf{C}$ is generated, with indices $i$ ranging from 1 to $m - k + 1$ and $j$ from 1 to $n$, through a sliding window technique applied to the tweets $e_{(i:i+k-1, j:j)}$:

$$c_{ij} = w_{ij} \cdot E_{e_{(i:i+k-1,j:j)}} + b_{ij}, \tag{3}$$

with $b_{ij}$ being a bias term.

The convolved features $\mathbf{C}$ are batch-normalized, activated by the Rectified Linear Unit (ReLU) function, and subsequently subjected to adaptive max pooling, resulting in an output size of $(1, 128)$. The pooled feature maps from different kernels sizes are concatenated to form $\mathbf{D} \in \mathbb{R}^{l \times 3 \times 128}$. The same process, employing a 2D-CNN with a kernel size of $(3, 1)$ and $l$ channels, is applied to generate $\mathbf{X} \in \mathbb{R}^{l \times 64}$.

$$\mathbf{X} = \text{Conv2d}(\mathbf{D}) \tag{4}$$

The dual-stage attention-based recurrent neural network is a widely recognized attentive RNN architecture that was originally introduced by Qin et al. [33] for time series forecasting. In our implementation, the Attentive LSTM is structured into two phases: the encoder phase consists of an input attention layer and an LSTM layer, followed by the decoder phase which includes a temporal attention layer and another LSTM layer, as illustrated in Fig. 1. The role of the input attention layer is to evaluate the significance of input features at a specific time instance $t$, and the temporal attention mechanism is applied in the decoder phase to adaptively select relevant encoder hidden states across all time steps.

For a multivariate time series, which is represented as $\mathbf{x}^k = (x_1^k, x_2^k, \ldots, x_l^k)^\top \in \mathbb{R}^l$, the formulation of an input attention mechanism can be achieved through the application of a deterministic attention model, specifically a Multilayer Perceptron (MLP). This process involves referencing the preceding hidden state $\mathbf{h}_{t-1}$ and the cell states $\mathbf{s}_{t-1}$ within the encoder LSTM unit, which is expressed in Eq. 5, where $\mathbf{v} \in \mathbb{R}^l$, $\mathbf{W}_e \in \mathbb{R}^{l \times 2z_1}$, and $\mathbf{U}_e \in \mathbb{R}^{l \times l}$ are parameters that are learned with $z_1$ being the size of hidden states. In the equation denoted as Eq. 6, the term $\alpha_t^k$ represents the attention weight, which signifies the importance of the $k$-th input feature at time $t$, where $n$ stands for the total number of features. To ensure that all attention weights sum up to 1, the softmax function is applied to $e_t^k$.

$$e_t^k = \mathbf{v}_e^\top \cdot \tanh\Big(\mathbf{W}_e \left[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}\right] + \mathbf{U}_e \mathbf{x}^k\Big) \tag{5}$$

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)} \tag{6}$$

The input attention mechanism is designed to be jointly trained with other components of the dual-stage Attentive LSTM. The hidden state at time $t$ is updated as follows:

$$\mathbf{h}_t = f_1(\mathbf{h}_{t-1}, \widetilde{\mathbf{x}}_t) \tag{7}$$

$$\widetilde{\mathbf{x}}_t = \left(\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \ldots, \alpha_t^n x_t^n\right)^\top \tag{8}$$

Here, $f$ represents a LSTM unit with $\mathbf{x}_t$ replaced by the newly computed $\widetilde{\mathbf{x}}_t$. The proposed input attention strategy enables the encoder to selectively focus on particular sequences of features rather than processing all input feature series uniformly. Following the encoding phase, the decoder equipped with a temporal attention mechanism is adopted to predict the output by leveraging LSTM to decode the encoded information. The temporal attention mechanism is designed to adaptively select relevant encoder hidden states in the decoder. Particularly, for each encoder hidden state at time $t$, the attention weight is determined by considering both the preceding decoder hidden state $\mathbf{h'}_{t-1} \in \mathbb{R}^{z_2}$ and the cell state of the LSTM unit $\mathbf{s'}_{t-1} \in \mathbb{R}^{z_2}$, where $z_2$ denotes the size of the decoder hidden states. This can be represented by Eq. 9 and Eq. 10.

$$d_t^j = \mathbf{v}_d^\top \cdot \tanh\left(\mathbf{W}_d \left[\mathbf{h'_{t-1}}; \mathbf{s'_{t-1}}\right] + \mathbf{U}_d \mathbf{h}_j\right), \quad 1 \le j \le l \tag{9}$$

$$\beta_t^j = \frac{\exp(d_t^j)}{\sum_1^l \exp(d_t^i)} \tag{10}$$

In this context, $[\mathbf{h'}_{t-1}; \mathbf{s'}_{t-1}] \in \mathbb{R}^{2z_2}$ is a concatenation of the previous hidden state and cell state of the LSTM unit. The parameters $\mathbf{v}_d \in \mathbb{R}^{z_1}$, $\mathbf{W}_d \in \mathbb{R}^{z_1 \times 2z_2}$, and $\mathbf{U}_d \in \mathbb{R}^{z_1 \times z_1}$ are to be learned with bias terms being excluded for clarity. For each encoder hidden state at time $t$, the attention weight $\beta_t^j$ represents the importance of the $i$-th encoder hidden state for the prediction and is calculated as shown in Eq. 10.

Subsequently, a context vector $\mathbf{c}_t$ is derived by calculating the weighted sum of encoder hidden states:

$$\mathbf{c}_t = \sum_{i=1}^{l} \beta_t^i \mathbf{h}_i \tag{11}$$

Once the weighted summed context vectors are obtained, they are combined with the given target series $(y_1, y_2, \ldots, y_{l-1})$ as follows:

$$\widetilde{y}_{t-1} = \widetilde{\mathbf{w}}^\top [y_{t-1}; \mathbf{c}_{t-1}] + \widetilde{b} \tag{12}$$

where $[y_{t-1}; \mathbf{c}_{t-1}] \in \mathbb{R}^{z_1+1}$ is a concatenation of the decoder input $y_{t-1}$ and the computed context vector $\mathbf{c}_{t-1}$. The parameters $\widetilde{\mathbf{w}} \in \mathbb{R}^{z_1+1}$ and $\widetilde{b} \in \mathbb{R}$ map the concatenation to the size of the decoder input. The newly computed $\tilde{y}_{t-1}$ is used to update the decoder hidden state at time $t$:

$$\mathbf{h}'_t = f_2(\mathbf{h'_{t-1}}, \tilde{y}_{t-1}) \tag{13}$$

This established attention strategy recognizes the variability in the information quality of tweets and their differential impacts across market phases.

### *E. Price Normalization*

Technical analysis indicates that historical price offers significant insights into prospective market movements [34]. The price normalization is designed to encapsulate the temporal sequence representation of quantitative indicators spanning a lookback period of $T$ days, which is 7 days in our study. The historical price data encompass open, high, low, close, and adjusted close prices for every trading session. The price movement indicators over intervals of 5, 10, 15, 20, 25 and 30 days are also computed. To capture the market fluctuations, we normalize both the price and its associated movement indicators employing the formula: $q_i = q_i / p_{i-1}^c$, where $q_i$ is the quantitative indicator and $p_{i-1}^c$ represents the previous session's adjusted close price. Consequently, a total of 11 normalized price indicators have been derived in this study [19]: open, high, low, close, adjusted close, 5 days, 10 days, 15 days, 20 days, 25 days and 30 days.

### *F. Price Encoder*

To capture the sequential interdependence in prices accross trading days, the same dual-stage attentive LSTM model is adopted. This process involves multivariate time series for quantitative indicators $\mathbf{q}^k = (q_1^k, q_2^k, \ldots, q_t^k) \in \mathbb{R}^l$ and defines encoder output on the $t^{th}$ day as follows:

$$\mathbf{p}_t = f_1(\mathbf{p}_{t-1}, \widetilde{\mathbf{q}}_t) \tag{14}$$

Here, $q_i \in \mathbb{R}^{11}$ denotes the price vector on day $i$ for each stock in the window of time steps.

Consistent with the perspective that each trading day's trend influences stock trend forecasting differently [13], we incorporate an identical temporal attention mechanism within the decoder to determine the importance of specific days, thereby creating a comprehensive feature representation from all hidden states of the LSTM [33]. This approach, similar to the implementation in the Tweets Encoder, leverages the newly computed $\tilde{y}_{t-1}$ to update the decoder's hidden state at time $t$ as follows:

$$\mathbf{p}'_t = f_2(\mathbf{p}'_{t-1}, \tilde{y}_{t-1}) \tag{15}$$

### G. Price Relationship Generator

In time series analysis, Dynamic Time Warping (DTW) is an effective algorithm for measuring the similarity between two one-dimensional temporal sequences [35]. Given stocks $A$ and $B$, which have time series of normalized price indicator $\mathbf{q}_a{}^k = (q^k_{a1}, q^k_{a2}, \ldots, q^k_{al})^\top \in \mathbb{R}^l$ and $\mathbf{q}_b{}^k = (q^k_{b1}, q^k_{b2}, \ldots, q^k_{bl})^\top \in \mathbb{R}^l$, the price relationship between $A$ and $B$ can be defined as:

$$r_q(A, B) = \overline{DTW(\mathbf{q}_a{}^k, \mathbf{q}_b{}^k)} \tag{16}$$

where we compute pairwise distances across all eleven quantitative indicators and subsequently synthesize them into a singular closeness metric through averaging.

### H. Graph Attention Networks

In the unified relation graph $\mathbb{G}(\mathbb{V}, \mathbb{E}, \mathbb{W})$, companies are represented by nodes $\mathbb{V}$ and edges $\mathbb{E}$, while the edge features $\mathbb{W}$ embody both the price and semantic relationships existing between nodes, derived from SRG and PRG. In instances where a relation is absent between any two given nodes, the corresponding edge feature value in the graph is assigned to zeros. This representation effectively captures the intricate network of interconnections and interactions among companies, delineated through both quantitative and textual dimensions. The relationship graph is subsequently passed to Graph Attention Network (GAT), representing the most popular architecture within the domain of Graph Neural Networks (GNNs), which are widely acknowledged for their cutting-edge capabilities in representation learning on graph-structured data [36]–[38]. In our study, we adopted GATv2, an advanced variant designed to address the limitations of static attention observed in the conventional GATConv layer. The standard GAT architecture suffers from a critical limitation where the sequential application of linear layers results in the ranking of attended nodes without considering the contextual relevance of the query node. GATv2, on the other hand, introduces a dynamic method wherein each node possesses the capability to attend to any other node in the network. This approach significantly enhances the adaptability and efficiency of the attention mechanism, allowing for a more nuanced and context-aware representation of node relationships within the graph.

Unlike other GNN architectures which assign equal importance to all neighboring nodes $j \in \mathcal{N}_i$ [39], GATs compute a learned weighted average of the representations of $\mathcal{N}_i$ given the layer's input of a set of node representations $\{\mathbf{g}_i \in \mathbb{R}^d | i \in \mathbb{V}\}$ and the set of edges $\mathbb{E}$ and layer's output of a new set of node representations $\{\mathbf{g}'_i \in \mathbb{R}^{d'} | i \in \mathbb{V}\}$. Specifically, GATs use a scoring function $o$ that maps $\mathbb{R}^d \times \mathbb{R}^d$ to $\mathbb{R}$ through computing a score for every edge $(j, i)$, which reflects the importance of the features of the neighboring node j with respect to the central node i, thereby enabling a differentiated and context-aware aggregation of neighborhood information.

$$o(\mathbf{g}_i, \mathbf{g}_j) = \mathbf{a}^\top \text{LeakyReLU}(\mathbf{W} \cdot [\mathbf{g}_i; \mathbf{g}_j]) \tag{17}$$

Here, $\mathbf{a} \in \mathbb{R}^{2d'}$, $\mathbf{W} \in \mathbb{R}^{d' \times d}$ are learned, and $\mathbf{g}_i$ and $\mathbf{g}_j$ are concatenated. The calculation of the attention scores involves a normalization step that is applied across the entire neighborhood $j \in \mathcal{N}_i$ using a softmax function:

$$\gamma_{ij} = \text{softmax}_j(o(\mathbf{g}_i, \mathbf{g}_j)) = \frac{\exp(o(\mathbf{g}_i, \mathbf{g}_j))}{\sum_{k \in \mathcal{N}_i} \exp(o(\mathbf{g}_i, \mathbf{g}_k))} \tag{18}$$

Subsequently, the Graph Attention Networks (GATs) ascertain the new representation of node $i$ by computing a weighted mean of the modified feature vectors of adjacent nodes. This computation incorporates a non-linear function $\sigma$ and utilizes the attention coefficients that have been normalized:

$$\mathbf{g}'_i = \sigma \left( \sum_{j \in \mathcal{N}_i} \gamma_{ij} \cdot \mathbf{W} \mathbf{g}_j \right) \tag{19}$$

TABLE I
DESCRIPTION OF "PRICE MOVEMENT - TEXT" DATASETS.

| | **CKIM18** | **BIGDATA22** |
|---|---|---|
| No. of stocks | 38 | 50 |
| No. of tweets | 955,788 | 272,762 |
| From | 2017-01-01 | 2019-07-05 |
| To | 2017-12-28 | 2020-06-30 |

### *I. Stock Price Movement Classifier*

The construction of the stock price movement classifier entails a multifaceted series of processes. Specifically, the feature representations derived from the Tweets Encoder, Price Encoder, and Graph Attention Networks, which are $\mathbf{T}$, $\mathbf{Q}$ and $\mathbf{H}$ respectively, are subjected to a systematic sequence of operations, which commences with a concatenation of different features, then proceeds to batch normalization. Subsequently, a linear layer equipped with a Rectified Linear Unit (ReLU) activation function is applied, followed by the introduction of a dropout layer, culminating in the deployment of a final linear layer dedicated to calculating the cross-entropy loss.

$$\mathbf{O} = \mathbf{W}_2 \cdot \text{Dropout}(\text{ReLU}(\mathbf{W}_1 \cdot \text{BatchNorm}([\mathbf{T};\mathbf{Q};\mathbf{H}]) + b_1)) + b_2 \tag{20}$$

where $\mathbf{W_1}$, $\mathbf{W_2}$, $b_1$ and $b_2$ are weights and bias terms to be optimized with cross entropy loss.

## IV. EXPERIMENTAL SETUP

### *A. Datasets*

In our study, we adopt the social text-driven stock prediction dataset developed by Wu et al. [18] (CIKM18), which combines stock prices sourced from Yahoo Finance and associated social media narratives, primarily from Twitter. This dataset spans from January 2017 to December 2017 and includes 38 stocks selected from the Standard & Poor's 500 index, each with significant Twitter presence. Additionally, we incorporate a recently released dataset for stock market forecasting, created by Soun et al. [19] (BIGDATA22), comprising 50 stocks and covering the period from July 2019 to June 2020. Relevant statistics of CIKM18 and BIGDATA22 can be viewed in Table I. We follow the binary classification task formulated in Section III-A and exclude data points which are not labeled as either 0 or 1. To evaluate our model, we split each dataset chronologically into training, validation, and test dataset, in line with the procedures used in the recent stock movement prediction research [19].

### *B. Evaluation Metrics*

To make fair comparison with previous studies on stock forecasting [13], [23], our chosen evaluation metrics are accuracy (ACC) and the Matthews Correlation Coefficient (MCC). ACC, a commonly utilized metric, is extensively adopted across various classification problems. The MCC becomes especially pertinent when the dataset presents notable disparities in class distribution. The calculation of the MCC necessitates the construction of a confusion matrix, which enumerates true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The formulas to compute ACC and MCC are defined as:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{21}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \tag{22}$$

Our findings represent the average performance of 10 iterations on the test sets, each with different random seeds.

### *C. Baseline Models*

We conduct a thorough comparison of our proposed model with established and robust baseline models in the context of forecasting stock price movements, as detailed subsequently:

**Logistic Regression (LR)** stands as a rudimentary benchmark, delineating a linear distinction amongst categories.

**Random Forest (RF)**, a potent attribute-driven method, frequently outperforms LR by amalgamating an assortment of randomized decision trees.

**StockNet** model is introduced by [23], where the input from the stock is processed using a Variational Autoencoder (VAE) to capture its inherent stochastic nature.

**Attentive LSTM (ALSTM)** integrates the attention mechanism with various LSTM cell states, as highlighted by [33]. Following this concept, two distinct variants, termed as ALSTM-W and ALSTM-D, have been proposed by [19]. Specifically, the ALSTM-W variant employs the Word2Vec technique to generate tweet embeddings, which are then incorporated into its predictive pipeline. Here, they focus on tweets that directly mention the targeted stocks and utilize Word2Vec to create embeddings for each tweet, compute their average, and subsequently merge this data with the price attribute in the ALSTM model. On the other hand, the ALSTM-D variant mirrors the ALSTM-W approach but leverages Doc2Vec for generating tweet embeddings. Although Doc2Vec provides a more comprehensive representation of textual content, it may sometimes fall short in terms of generalization performance.

**Attentive LSTM using adversarial training (Adv-ALSTM)** is introduced by [13], where the model enhances the traditional ALSTM by integrating adversarial training. This integration aims to improve the model's ability to generalize across different scenarios.

**DTML**, as proposed by [40], represents a novel methodology for the accurate prediction of stock movements. This approach is distinguished by its ability to establish effective correlations among multiple stocks. DTML is designed to leverage both temporal and global market contexts, enabling it to dynamically comprehend inter-stock relationships. Its performance exceeds that of existing methods.

**SLOT**, introduced by [19], enhances the prediction of stock market trends by utilizing self-supervised learning techniques applied to the sparse and often noisy data available from tweets. It effectively mitigates the prevalent biases associated with highly popular stocks and efficiently excludes irrelevant data. SLOT involves the creation of shared embeddings for both stocks and tweets through self-supervised learning, which is particularly beneficial for generating accurate predictions for stocks with lower popularity. Furthermore, the method is designed to exploit multi-level relationships among stocks as deduced from tweet data, thereby significantly bolstering its predictive robustness.

### *D. Experimental Details*

In this study, the models are trained on a computational infrastructure equipped with an NVIDIA Tesla T4 processor. The training procedure spanned a total of 50 epochs. Within the training process, the validation dataset is utilized to select the most optimal model, while the test dataset is employed to report the model performance. Furthermore, a learning rate of 1e-6 is applied in tandem with the Adam optimizer. The selection of these particular configurations and hyperparameters has been undertaken with meticulous consideration, with the objective of ensuring optimal model convergence and robustness throughout the training and evaluation stages.

## V. Result and Analysis

Our model demonstrates superior performance across various categories and evaluation metrics in Table II, outperforming traditional technical analysis methods [41] and machine learning approaches [13], [19], [23], [33], [40] by significant margins. Specifically, the model achieved the highest levels of accuracy on the CIKM18 and BIGDATA22 datasets, recording accuracy scores of 0.5730 and 0.5833, respectively, which reflect improvements of 2.5% and 6.4% over the strongest baselines. Additionally, the model attained its peak MCC score of 0.1556 on the BIGDATA22 dataset and the second-highest MCC score of 0.0389 on the CIKM18 datasets. On average, our model surpasses the most competitive baseline models by 4.5% in accuracy across these datasets.

The improvements confirm the effectiveness of our proposed framework. In the following ablation study, we will test the utility of the proposed dynamic relationship modeling methods for text and prices.

## VI. Ablation Study

An ablation study was performed to evaluate the effectiveness of PRG and SRG in predicting stock price movements. The results, detailed in Table III highlight the positive impacts of the PRG and SRG modules on the models' performance and stability.

Specifically, the inclusion of the SRG module markedly improved model performance, increasing accuracy scores from 0.5551 to 0.5650 for the CIKM18 dataset and from 0.5609 to 0.5689 for the BIGDATA22 dataset, in comparison to the baseline model which only incorporated the Price Encoder (PE) and Tweets Encoder (TE). Similarly, the integration of PRG resulted in enhanced performance, elevating scores from 0.5551 to 0.5661 for CIKM18 and from 0.5609 to 0.5712 for BIGDATA22. PRG, focusing directly on price movements, showed a more pronounced improvement in performance compared to SRG.

Our view is that this is primarily attributed to the more apparent relationships in quantitative data as opposed to the sparse and varied content found in tweets. Furthermore, the combined use of both SRG and PRG led to a significant increase in model accuracy by approximately 2 percentage points and an improvement in MCC by 2% to 7%, which highlights the critical role of price relationship information in forecasting, with semantic relationship information assuming a supplementary role.

TABLE II
PERFORMANCE COMPARISON ON THE BENCHMARK DATASET. THE BOLDFACE INDICATES THE HIGHEST SCORES.

| **Model** | **CIKM18** | | **BIGDATA22** | |
|---|---|---|---|---|
| | Accuracy | MCC | Accuracy | MCC |
| LR [40] | 0.5250 | -0.0425 | 0.5307 | 0.0200 |
| RF [40] | 0.5357 | 0.0119 | 0.4710 | -0.1114 |
| ALSTM [33] | 0.5254 | -0.0077 | 0.4869 | -0.0254 |
| ALSTM-W [19] | 0.5364 | 0.0315 | 0.4828 | -0.0116 |
| ALSTM-D [19] | 0.5040 | -0.0449 | 0.4916 | -0.0090 |
| Adv-ALSTM [13] | 0.5369 | 0.0217 | 0.5036 | 0.0120 |
| StockNet [23] | 0.5235 | -0.0161 | 0.5299 | -0.0163 |
| DTML [40] | 0.5386 | 0.0049 | 0.5165 | 0.0651 |
| SLOT [19] | 0.5586 | **0.0899** | 0.5481 | 0.0952 |
| **Ours** | **0.5730** | 0.0389 | **0.5833** | **0.1556** |

TABLE III
ABLATION ANALYSIS FOR STOCK PRICE MOVEMENT PREDICTION USING DYNAMIC DUAL-GRAPH NEURAL NETWORK. BOLDFACE INDICATED THE BEST RESULT.

| **Model** | **CIKM18** | | **BIGDATA22** | |
|---|---|---|---|---|
| | Accuracy | MCC | Accuracy | MCC |
| TE + PE | 0.5551 | 0.0197 | 0.5609 | 0.0898 |
| TE + PE + SRG | 0.5650 | 0.0289 | 0.5689 | 0.1391 |
| TE + PE + PRG | 0.5661 | 0.0314 | 0.5712 | 0.1476 |
| **TE + PE + PRG + SRG** | **0.5730** | **0.0389** | **0.5833** | **0.1556** |

## VII. CONCLUSION

A dynamic dual-graph neural network architecture is proposed, aiming to leverage implicit relationships between stocks to yield more accurate stock price movement predictions. The network simultaneously encodes price and textual information of companies, and builds two dynamic graphs using quantitative and semantic relations. A distinctive feature of our architecture is the adoption of Attentive LSTM in conjunction with Graph Attention Networks, which is designed to unravel the underlying connections between stock prices and textual information. We also conducted extensive experiments on two benchmark datasets for stock movement prediction, which exhibit superior performances compared to other graph-based stock market prediction models.

The improvements in our method over strong baselines demonstrate the effectiveness of our stock correlation modeling method. In future work, we will leverage explainable AI [42], [43] to investigate the correlations and causalities [44] of company relationships and their impact on stock price movement.

## REFERENCES


[1] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, "Text mining of news-headlines for forex market prediction: A multi-layer dimension reduction algorithm with semantics and sentiment," *Expert Systems with Applications*, vol. 42, no. 1, pp. 306–324, 2015.

[2] L. Malandri, F. Xing, C. Orsenigo, C. Vercellis, and E. Cambria, "Public mood–driven asset allocation: the importance of financial sentiment in portfolio management," *Cognitive Computation*, vol. 10, no. 6, pp. 1167–1176, 2018.

[3] A. Mudinas, D. Zhang, and M. Levene, "Market trend prediction using sentiment analysis: lessons learned and paths forward," *arXiv preprint arXiv:1903.05440*, 2019.

[4] F. Xing, L. Malandri, Y. Zhang, and E. Cambria, "Financial sentiment analysis: An investigation into common mistakes and silver bullets," in *COLING*, 2020, pp. 978–987.

[5] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, "Multi-source aggregated classification for stock price movement prediction." *Information Fusion*, vol. 91, pp. 515–528, 2023.

[6] W. J. Yeo, W. van der Heever, R. Mao, E. Cambria, R. Satapathy, and G. Mengaldo, "A comprehensive review on financial explainable AI," *arXiv preprint arXiv:2309.11960*, 2023.

[7] Y. Ma, R. Mao, Q. Lin, P. Wu, and E. Cambria, "Quantitative stock portfolio optimization by multi-task learning risk and return," *Information Fusion*, vol. 104, p. 102165, 2024.

[8] K. Du, F. Xing, and E. Cambria, "Incorporating multiple knowledge sources for targeted aspect-based financial sentiment analysis," *ACM Transactions on Management Information Systems*, vol. 14, no. 3, p. 23, 2023.

[9] L. Luo, S. You, Y. Xu, and H. Peng, "Improving the integration of piece wise linear representation and weighted support vector machine for stock trading signal prediction," *Applied soft computing*, vol. 56, pp. 199–216, 2017.

[10] Y.-H. Chang and M.-S. Lee, "Incorporating markov decision process on genetic algorithms to formulate trading strategies for stock markets," *Applied Soft Computing*, vol. 52, pp. 1143–1153, 2017.

[11] F. Jin, W. Wang, P. Chakraborty, N. Self, F. Chen, and N. Ramakrishnan, "Tracking multiple social media for stock market event prediction," in *Advances in Data Mining. Applications and Theoretical Aspects: 17th Industrial Conference, ICDM 2017, New York, NY, USA, July 12-13, 2017, Proceedings 17*. Springer, 2017, pp. 16–30.

[12] J. Chang and W. Tu, "A stock-movement aware approach for discovering investors' personalized preferences in stock markets," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2018, pp. 275–280.

[13] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T. Chua, "Enhancing stock movement prediction with adversarial training," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 5843–5849.

[14] H. Liu and B. Song, "Stock trends forecasting by multi-layer stochastic ann bagging," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017, pp. 322–329.

[15] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 2141–2149.

[16] C. Chen, L. Zhao, J. Bian, C. Xing, and T.-Y. Liu, "Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2376–2384.

[17] J. Ye, J. Zhao, K. Ye, and C. Xu, "Multi-graph convolutional network for relationship-driven stock movement prediction," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 6702–6709.

[18] H. Wu, W. Zhang, W. Shen, and J. Wang, "Hybrid deep sequential modeling for social text-driven stock prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, 2018, pp. 1627–1630.

[19] Y. Soun, J. Yoo, M. Cho, J. Jeon, and U. Kang, "Accurate stock movement prediction with self-supervised learning from sparse noisy tweets," in *IEEE International Conference on Big Data*, 2022, pp. 1691–1700.

[20] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 2327–2333.

[21] Z. Zhou, L. Ma, and H. Liu, "Trade the event: Corporate events detection for news-based event-driven trading," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 2114–2124.

[22] Y. Liu, Q. Zeng, H. Yang, and A. Carrio, "Stock price movement prediction from financial news with deep learning and knowledge graph embedding," in *Pacific rim Knowledge Acquisition Workshop*. Springer, 2018, pp. 102–113.

[23] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1970–1979.

[24] E. F. Fama, "The behavior of stock-market prices," *The journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.

[25] Y. Chen, Z. Wei, and X. Huang, "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*, 2018, pp. 1655–1658.

[26] R. Sawhney, S. Agarwal, A. Wadhwa, and R. Shah, "Deep attentive learning for stock movement prediction from social media text and company correlations," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8415–8426.

[27] R. Mao, K. Du, Y. Ma, L. Zhu, and E. Cambria, "Discovering the cognition behind language: Financial metaphor analysis with MetaPro," in *2023 IEEE International Conference on Data Mining*, 2023.

[28] F. Xing, E. Cambria, and R. Welsch, "Intelligent asset allocation via market sentiment views," *IEEE Computational Intelligence Magazine*, vol. 13, no. 4, pp. 25–34, 2018.

[29] R. Mao, X. Li, M. Ge, and E. Cambria, "MetaPro: A computational metaphor processing model for text pre-processing," *Information Fusion*, vol. 86-87, pp. 30–43, 2022.

[30] R. Mao, X. Li, K. He, M. Ge, and E. Cambria, "MetaPro Online: A computational metaphor processing online system," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Toronto, Canada: Association for Computational Linguistics, 2023, pp. 127–135. [Online]. Available: https://aclanthology.org/2023.acl-demo.12

[31] K. Du, F. Xing, R. Mao, and E. Cambria, "Financial sentiment analysis: Techniques and applications," *ACM Computing Surveys*, 2024.

[32] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-Networks," in *Conference on Empirical Methods in Natural Language Processing*, 2019.

[33] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2627–2633.

[34] M. Jeanblanc, M. Yor, and M. Chesney, *Mathematical methods for financial markets*. Springer Science & Business Media, 2009.

[35] K. Wang and T. Gasser, "Alignment of curves by dynamic time warping," *The Annals of Statistics*, vol. 25, no. 3, 1997.

[36] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[37] Q. Lin, J. Liu, R. Mao, F. Xu, and E. Cambria, "TECHS: Temporal logical graph networks for explainable extrapolation reasoning," in *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL)*, vol. 1, 2023, pp. 1281–1293.

[38] Q. Lin, R. Mao, J. Liu, F. Xu, and E. Cambria, "Fusing topology contexts and logical rules in language models for knowledge graph completion," *Information Fusion*, vol. 90, pp. 253–264, 2023.

[39] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[40] J. Yoo, Y. Soun, Y.-c. Park, and U. Kang, "Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2037–2045.

[41] E. Chan, *Algorithmic Trading: Winning Strategies and Their Rationale*. Wiley, 2013.

[42] S. Han, R. Mao, and E. Cambria, "Hierarchical attention network for explainable depression detection on Twitter aided by metaphor concept mappings," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 94–104.

[43] H. Turbé, M. Bjelogrlic, C. Lovis, and G. Mengaldo, "Evaluation of post-hoc interpretability methods in time-series classification," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 250–260, 2023.

[44] E. Cambria, R. Mao, M. Chen, Z. Wang, and S.-B. Ho, "Seven pillars for the future of artificial intelligence," *IEEE Intelligent Systems*, vol. 38, no. 6, pp. 62–69, 2023.