DGDNN: Decoupled Graph Diffusion Neural Network for Stock Movement Prediction

Zinuo You[1], Zijian Shi[1], Hongbo Bo[1,3], John Cartlidge[2], Li Zhang[4] and Yan Ge[2]

[1]*School of Computer Science, University of Bristol, Bristol, U.K.*

[2]*School of Engineering Mathematics and Technology, University of Bristol, Bristol, U.K.*

[3]*NIHR Innovation Observatory, Population Health Sciences Institute, Newcastle University, Newcastle, U.K.*

[4]*Department of Engineering Science, University of Oxford, U.K.*

Keywords: Stock Prediction, Graph Neural Network, Graph Structure Learning, Information Propagation.

Abstract: Forecasting future stock trends remains challenging for academia and industry due to stochastic inter-stock dynamics and hierarchical intra-stock dynamics influencing stock prices. In recent years, graph neural networks have achieved remarkable performance in this problem by formulating multiple stocks as graph-structured data. However, most of these approaches rely on artificially defined factors to construct static stock graphs, which fail to capture the intrinsic interdependencies between stocks that rapidly evolve. In addition, these methods often ignore the hierarchical features of the stocks and lose distinctive information within. In this work, we propose a novel graph learning approach implemented without expert knowledge to address these issues. First, our approach automatically constructs dynamic stock graphs by entropy-driven edge generation from a signal processing perspective. Then, we further learn task-optimal dependencies between stocks via a generalized graph diffusion process on constructed stock graphs. Last, a decoupled representation learning scheme is adopted to capture distinctive hierarchical intra-stock features. Experimental results demonstrate substantial improvements over state-of-the-art baselines on real-world datasets. Moreover, the ablation study and sensitivity study further illustrate the effectiveness of the proposed method in modeling the time-evolving inter-stock and intra-stock dynamics.

1 INTRODUCTION

The stock market has long been an intensively discussed research topic by investors pursuing profitable trading opportunities and policymakers attempting to gain market insights. Recent research advancements have primarily concentrated on exploring the potential of deep learning models, driven by their ability to model complex non-linear relationships (Bo et al., 2023) and automatically extract high-level features from raw data (Akita et al., 2016; Shi and Cartlidge, 2022). These abilities further enable the capture of intricate patterns in stock market data that traditional statistical methods might omit. However, the efficient market theory (Malkiel, 2003) and the random walk nature of stock prices make it challenging to predict exact future prices with high accuracy (Adam et al., 2016). As a result, research efforts have shifted towards the more robust task of anticipating stock movements (Jiang, 2021).

Early works (Roondiwala et al., 2017; Bao et al., 2017) commonly adopt deep learning techniques to extract temporal features from historical stock data and predict stock movements accordingly. However, these methods assume independence between stocks, neglecting their rich connections. In reality, stocks are often interrelated from which valuable information can be derived. These complicated relations between stocks are crucial for understanding the stock markets (Deng et al., 2019; Feng et al., 2019b; Feng et al., 2022).

To bridge this gap, some deep learning models attempt to model the interconnections between stocks by integrating textual data (Sawhney et al., 2020), such as tweets (Xu and Cohen, 2018) and news (Li et al., 2020b). Nevertheless, these models heavily rely on the quality of embedded extra information, resulting in highly volatile performance. Meanwhile, the transformer-based methods introduce different attention mechanisms to capture inter-stock relations based on multiple time series (i.e., time series of stock indicators, such as open price, close price, highest price,

You, Z., Shi, Z., Bo, H., Cartlidge, J., Zhang, L. and Ge, Y.
DGDNN: Decoupled Graph Diffusion Neural Network for Stock Movement Prediction.
DOI: 10.5220/0012406400003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 2*, pages 431-442
ISBN: 978-989-758-680-4; ISSN: 2184-433X

lowest price, and trading volume) (Yoo et al., 2021; Ding et al., 2021). Despite this advancement, these methods often lack explicit modeling of temporal information of these time series, such as temporal order and inter-series information (Zhu et al., 2021; Wen et al., 2022).

Recently, Graph Neural Networks (GNNs) have shown promising performance in analyzing various real-world networks or systems by formulating them as graph-structured data, such as transaction networks (Pareja et al., 2020), traffic networks (Wang et al., 2020), and communication networks (Li et al., 2020a). Typically, these networks possess multiple entities interacting over time, and time series data can characterize each entity. Analyzing stock markets as complex networks is a natural choice, as previous works indicate (Liu and Arunkumar, 2019; Shahzad et al., 2018). Moreover, various interactive mechanisms (e.g., transmitters and receivers (Shahzad et al., 2018)) that exist between stocks can be easily represented by edges (Cont and Bouchaud, 2000). Therefore, these properties make GNNs powerful candidates for explicitly grasping inter-stock relations and capturing intra-stock patterns with stock graphs (Sawhney et al., 2021a; Xiang et al., 2022).

However, existing GNN-based models face two fundamental challenges for stock movement prediction: representing complicated time-evolving inter-stock dependencies and capturing hierarchical features of stocks. First, specific groups of related stocks are affected by various factors, which change stochastically over time (Huynh et al., 2023). Most graph-based models (Kim et al., 2019; Ye et al., 2021; Sawhney et al., 2021b) construct time-invariant stock graphs, which are contrary to the stochastic and time-evolving nature of the stock market (Adam et al., 2016). For instance, inter-stock relations are commonly pre-determined by sector or firm-specific relationships (e.g., belonging to the same industry (Sawhney et al., 2021b) or sharing the same CEO (Kim et al., 2019)). Besides, artificially defined graphs for specific tasks may not be versatile or applicable to other tasks. Sticking to rigid graphs risks introducing noise and task-irrelevant patterns to models (Chen et al., 2020). Therefore, generating appropriate stock graphs and learning task-relevant topology remains a preliminary yet critical part of GNN-based methods in predicting stock movements. Second, stocks possess distinctive hierarchical features (Mantegna, 1999; Sawhney et al., 2021b) that remain under-exploited (e.g., overall market trends, group-specific dynamics, and individual trading patterns (Huynh et al., 2023)). Previous works indicate that these hierarchical intra-stock features could distinguish highly related stocks from different levels and be utilized for learning better and more robust representations (Huynh et al., 2023; Mantegna, 1999). However, in the conventional GNN-based methods, representation learning is combined with the message-passing process between immediate neighbors in the Euclidean space. As a result, node representations become overly similar as the message passes, severely distorting the distinctive individual node information (Huang et al., 2020; Rusch et al., 2023; Liu et al., 2020). Hence, preserving these hierarchical intra-stock features is necessary for GNN-based methods in predicting stock movements.

In this paper, we propose the Decoupled Graph Diffusion Neural Network (DGDNN) to address the abovementioned challenges. Overall, we treat stock movement prediction as a temporal node classification task and optimize the model toward identifying movements (classes) of stocks (nodes) on the next trading day. The main contributions of this paper are summarised as follows:

- We exploit the information entropy of nodes as their pair-wise connectivities with ratios of node energy as weights, enabling the modeling of intrinsic time-varying relations between stocks from the view of information propagation.
- We extend the layer-wise update rule of conventional GNNs to a decoupled graph diffusion process. This allows for learning the task-optimal graph topology and capturing the hierarchical features of multiple stocks.
- We conduct extensive experiments on real-world stock datasets with 2,893 stocks from three markets (NASDAQ, NYSE, and SSE). The experimental results demonstrate that DGDNN significantly outperforms state-of-the-art baselines in predicting the next trading day movement, with improvements of 9.06% in classification accuracy, 0.09 in Matthew correlation coefficient, and 0.06 in F1-Score.

2 RELATED WORK

This section provides a brief overview of relevant studies.

2.1 GNN-Based Methods for Modeling Multiple Stocks

The major advantage of applying GNNs lies in their graphical structure, which allows for explicitly modeling the relations between entities. For instance, STHAN-SR (Sawhney et al., 2021a), which is similar to the Graph Attention Neural Networks (GATs) (Veličković et al., 2018), adopts a spatial-temporal attention mechanism on a hypergraph with industry and corporate edges to capture inter-stock relations on the temporal domain and spatial domain. HATS (Kim et al., 2019) predicts the stock movement by a GAT-based method that the immediate neighbor nodes are selectively aggregated with learned weights on manually crafted multi-relational stock graphs. Moreover, HyperStockGAT (Sawhney et al., 2021b) leverages graph learning in hyperbolic space to capture the heterogeneity of node degree and hierarchical nature of stocks on an industry-related stock graph. This method illustrates that the node degree of stock graphs is not evenly distributed. Nonetheless, these methods directly correlate the stocks by empirical assumptions or expert knowledge to construct static stock graphs, contradicting the time-varying nature of the stock market.

2.2 Graph Topology Learning

To address the constraint of GNNs relying on the quality of raw graphs, researchers have proposed graph structure learning to optimize raw graphs for improved performance in downstream tasks. These methods can be broadly categorized into direct parameterizing approaches and neural network approaches. In the former category, methods treat the adjacency matrix of the target graph as free parameters to learn. Pro-GNN, for instance, demonstrates that refined graphs can gain robustness by learning perturbed raw graphs guided by critical properties of raw graphs (Jin et al., 2020). GLNN (Gao et al., 2020) integrates sparsity, feature smoothness, and initial connectivity into an objective function to obtain target graphs. In contrast, neural network approaches employ more complex neural networks to model edge weights based on node features and representations. For example, SLCNN utilizes two types of convolutional neural networks to learn the graph structure at both the global and local levels (Zhang et al., 2020). GLCN integrates graph learning and convolutional neural networks to discover the optimal graph structure that best serves downstream tasks (Jiang et al., 2019). Despite these advancements, direct parameterizing approaches often necessitate complex and time-consuming alternating optimizations or bi-level optimizations, and neural network approaches may overlook the unique characteristics of graph data or lose the positional information of nodes.

2.3 Decoupled Representation Learning

Various networks or systems exhibit unique characteristics that are challenging to capture within the constraints of Euclidean space, particularly when relying on manually assumed prior knowledge (Huynh et al., 2023; Sawhney et al., 2021b). In addressing this challenge, DAGNN (Liu et al., 2020) offers theoretical insights, emphasizing that the entanglement between representation transformation and message propagation can hinder the performance of message-passing GNNs. SHADOW-GNN (Zeng et al., 2021), on the other hand, concentrates on decoupling the representation learning process both in depth and scope. By learning on multiple subgraphs with arbitrary depth, SHADOW-GNN preserves the distinctive information of localized subgraphs instead of globally smoothing them into white noise. Another approach, MMP (Chen et al., 2022), transforms updated node messages into self-embedded representations. It then selectively aggregates these representations to form the final graph representation, deviating from the direct use of representations from the message-passing process.

3 PRELIMINARY

In this section, we present the fundamental notations used throughout this paper and details of the problem setting. Nodes represent stocks, node features represent their historical stock indicators, and edges represent interconnections between stocks.

3.1 Notation

Let $\mathcal{G}_t(\mathcal{V},\mathcal{E}_t)$ represents a weighted and directed graph on trading day t, where $\mathcal{V}$ is the set of nodes (stocks) $\{v_1,...,v_N\}$ with the number of nodes as $|\mathcal{V}| = N$, and $\mathcal{E}_t$ is the set of edges (inter-stock relations). Let $\mathbf{A}_t \in \mathbb{R}^{N\times N}$ represents the adjacency matrix and its entry $(\mathbf{A}_t)_{i,j}$ represents an edge from v_i to v_j. The node feature matrix is denoted as $\mathbf{X}_t \in \mathbb{R}^{N\times(\tau M)}$, where M represents the number of stock indicators (i.e., open price, close price, highest price, lowest price, trading volume, etc), and τ represents the length of the historical lookback window. The feature vector of v_i on trading day t is denoted as $\mathbf{x}_{t,i}$. Let $c_{t,i}$ represent the label of v_i on trading day t, where $C_t \in \mathbb{R}^{N\times 1}$ is the label matrix on trading day t.

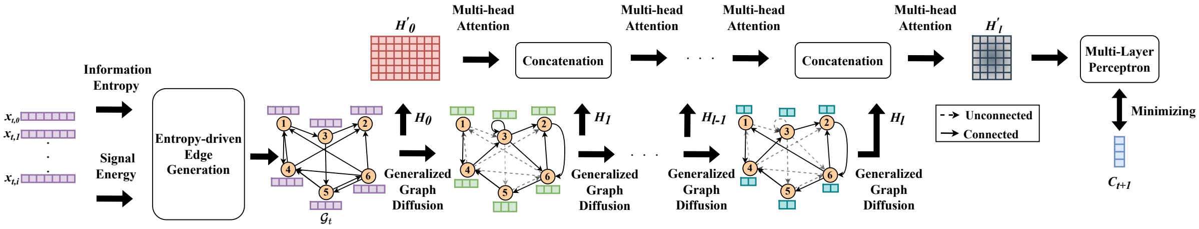


Figure 1: The DGDNN framework consists of three steps: (1) constructing the raw stock graph $\mathcal{G}_t$ (see Section 4.1); (2) learning the task-optimal graph topology by generalized graph diffusion (see Section 4.2); (3) applying a hierarchical decoupled representation learning scheme (see Section 4.3).

3.2 Problem Setting

Since we are predicting future trends of multiple stocks by utilizing their corresponding stock indicators, we transform the regression task of predicting exact stock prices into a temporal node classification task. Similar to previous works on stock movement prediction (Kim et al., 2019; Xiang et al., 2022; Sawhney et al., 2021a; Xu and Cohen, 2018; Li et al., 2021), we refer to this common and important task as next trading day stock trend classification. Given a set of stocks on the trading day t, the model learns from a historical lookback window of length τ (i.e., $[t-\tau+1, t]$) and predicts their labels in the next timestamp (i.e., trading day $t+1$). The mapping relationship of this work is expressed as follows,

$$f(\mathcal{G}_t(\mathcal{V}, \mathcal{E}_t)) \rightarrow C_{t+1}. \quad (1)$$

Here, $f(\cdot)$ represents the proposed method DGDNN.

4 METHODOLOGY

In this section, we detail the framework of the proposed DGDNN in depth, as depicted in Fig 1.

4.1 Entropy-Driven Edge Generation

Defining the graph structure is crucial for achieving reasonable performance for GNN-based approaches. In terms of stock graphs, traditional methods often establish static relations between stocks through human labeling or natural language processing techniques. However, recent practices have proven that generating dynamic relations based on historical stock indicators is more effective (Li et al., 2021; Xiang et al., 2022). These indicators, as suggested by previous financial studies (Dessaint et al., 2019; Cont and Bouchaud, 2000; Liu and Arunkumar, 2019), can be treated as noisy temporal signals. Simultaneously, stocks can be viewed as transmitters or receivers of information signals, influencing other stocks (Shahzad et al., 2018; Ferrer et al., 2018). Additionally, stock markets exhibit significant node-degree heterogeneity, with highly influential stocks having relatively large node degrees (Sawhney et al., 2021b; Arora et al., 2006).

Consequently, we propose to model interdependencies between stocks by treating the stock market as a communication network. Prior research (Yue et al., 2020) generates the asymmetric inter-stock relations based on transfer entropy. Nonetheless, the complex estimation process of transfer entropy and the limited consideration of edge weights hamper the approximation of the intrinsic inter-stock connections.

To this end, we quantify the links between nodes by utilizing the information entropy as the directional connectivity and signal energy as its intensity. On the one hand, if the information can propagate between entities within real-world systems, the uncertainty or randomness is reduced, resulting in a decrease in entropy and an increase in predictability at the receiving entities (Jaynes, 1957; Csiszár et al., 2004). On the other hand, the energy of the signals reflects their intensity during propagation, which can influence the received information at the receiver. The entry $(\mathbf{A}_t)_{i,j}$ is defined by,

$$(\mathbf{A}_t)_{i,j} = \frac{E(\mathbf{x}_{t,i})}{E(\mathbf{x}_{t,j})}\left(e^{S(\mathbf{x}_{t,i})+S(\mathbf{x}_{t,j})-S(\mathbf{x}_{t,i},\mathbf{x}_{t,j})} - 1\right). \quad (2)$$

Here, $E(\cdot)$ denotes the signal energy, and $S(\cdot)$ denotes the information entropy. The signal energy of v_i is obtained by,

$$E(\mathbf{x}_{t,i}) = \sum_{n=0}^{\tau M-1} |\mathbf{x}_{t,i}[n]|^2. \quad (3)$$

The information entropy of v_i is obtained by,

$$S(\mathbf{x}_{t,i}) = -\sum_{j=0} p(s_j) \ln p(s_j), \quad (4)$$

where $\{s_0, \ldots, s_j\}$ denotes the non-repeating sequence of $\mathbf{x}_{t,i}$ and $p(s_j)$ represents the probability of value s_j. By definition, we can obtain $p(s_j)$ by,

$$p(s_j) = \frac{\sum_{n=0}^{\tau M-1} \delta(s_j - \mathbf{x}_{t,i}[n])}{\tau M}. \quad (5)$$

Here $\delta(\cdot)$ denotes the Dirac delta function.

4.2 Generalized Graph Diffusion

However, simply assuming constructed graphs are perfect for performing specific tasks can lead to discordance between given graphs and task objectives, resulting in sub-optimal model performance (Chen et al., 2020). Several methods have been proposed to mitigate this issue, including AdaEdge (Chen et al., 2020) and DropEdge (Rong et al., 2019). These methods demonstrate notable improvements in node classification tasks by adding or removing edges to perturb graph topologies, enabling models to capture and leverage critical topological information.

With this in mind, we propose to utilize a generalized diffusion process on the constructed stock graph to learn the task-optimal topology. It enables more effective capture of long-range dependencies and global information on the graph by diffusing information across larger neighborhoods (Klicpera et al., 2019).

The following equation defines the generalized graph diffusion at layer l,

$$\mathbf{Q}_l = \sum_{k=0}^{K-1} \theta_{l,k}\mathbf{T}_{l,k}, \; \sum_{k=0}^{K-1} \theta_{l,k} = 1. \tag{6}$$

Here $\mathbf{Q}_l$ denotes the diffusion matrix, K denotes the maximum diffusion step, $\theta_{l,k}$ denotes the weight coefficients, and $\mathbf{T}_{l,k}$ denotes the column-stochastic transition matrix. Specifically, generalized graph diffusion transforms the given graph structure into a new one while keeping node signals neither amplified nor reduced. Consequently, the generalized graph diffusion turns the information exchange solely between adjacent connected nodes into broader unconnected areas of the graph.

Notably, $\theta_{l,k}$ and $\mathbf{T}_{l,k}$ can be determined in advance (Klicpera et al., 2019). For instance, we can use the heat kernel or the personalized PageRank to define $\theta_{l,k}$, and the random walk transition matrix or symmetric transition matrix to define $\mathbf{T}_{l,k}$. Although these pre-defined mappings perform well in some datasets (e.g., CORA, CiteSeer, and PubMed) with time-invariant relations (Zhao et al., 2021), they are not feasible for tasks that require considering changing relationships.

Therefore, we make $\theta_{l,k}$ as trainable parameters, $\mathbf{T}_{l,k}$ as trainable matrices, and K as a hyperparameter. Furthermore, we introduce a neighborhood radius (Zhao et al., 2021) to control the effectiveness of the generalized graph diffusion. The neighborhood radius at layer l is expressed as,

$$r_l = \frac{\sum_{k=0}^{K-1} \theta_{l,k} k}{\sum_{k=0}^{K-1} \theta_{l,k}}, \; r_l > 0 \tag{7}$$

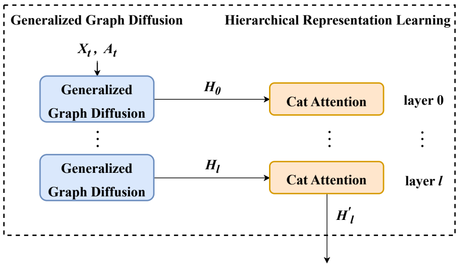

Generalized Graph Diffusion
Hierarchical Representation Learning
X_t, A_t
Generalized Graph Diffusion
H_0
Cat Attention
layer 0
Generalized Graph Diffusion
H_l
Cat Attention
layer l
H'_l

Figure 2: The component-wise layout of hierarchical decoupled representation learning with input $\mathbf{X}_t$, $\mathbf{A}_t$.

Here, large r_l indicates the model explores more on distant nodes and vice versa.

4.3 Hierarchical Decoupled Representation Learning

Theoretically, GNNs update nodes by continuously aggregating direct one-hop neighbors, producing the final representation. However, this can lead to a high distortion of the learned representation. It is probably because the message-passing and representation transformation do not essentially share a fixed neighborhood in the Euclidean space (Liu et al., 2020; Xu et al., 2018; Chen et al., 2020). To address this issue, decoupled GNNs have been proposed (Liu et al., 2020; Xu et al., 2018), aiming to decouple these two processes and prevent the loss of distinctive local information in learned representation. Similarly, methods such as HyperStockGAT (Sawhney et al., 2021b) have explored learning graph representations in hyperbolic spaces with attention mechanisms to capture temporal features of stocks at different levels.

Inspired by these methods, we adopt a hierarchical decoupled representation learning strategy to capture hierarchical intra-stock features. Each layer in DGDNN comprises a Generalized Graph Diffusion layer and a Cat Attention layer in parallel, as depicted in Fig. 2. The layer-wise update rule is defined by,

$$\begin{aligned} \mathbf{H}_l &= \sigma\left((\mathbf{Q}_l \odot \mathbf{A}_t)\mathbf{H}_{l-1}\mathbf{W}_l^0\right), \\ \mathbf{H}'_l &= \sigma\left(\zeta(\mathbf{H}_l || \mathbf{H}'_{l-1})\mathbf{W}_l^1 + b_l^1\right). \end{aligned} \tag{8}$$

Here, $\mathbf{H}'_l$ denote the node representation of $l-th$ layer, $\sigma(\cdot)$ is the activation function, $\zeta(\cdot)$ denotes the multi-head attention, $||$ denotes the concatenation, and $\mathbf{W}_l$ denotes the layer-wise trainable weight matrix.

4.4 Objective Function

According to Eq. 1, Eq. 6, and Eq. 7, we formulated the objective function of DGDNN as follows,

$$\mathcal{J} = \frac{1}{B}\sum_{t=0}^{B-1}\mathcal{L}_{CE}(C_{t+1}, f(\mathbf{X}_t, \mathbf{A}_t)) - \alpha\sum_{l=0}^{L-1} r_l + \sum_{l=0}^{L-1}(\sum_{k=0}^{K-1}\theta_{l,k} - 1). \tag{9}$$

Here, $\mathcal{L}_{CE}(\cdot)$ denotes the cross-entropy loss, B denotes the batch size, L denotes the number of information propagation layers, and α denotes the weight coefficient controlling the neighborhood radius.

5 EXPERIMENT

The experiments are conducted on 3x Nvidia Tesla T4, CUDA version 11.2. Datasets and source code are available[1].

5.1 Dataset

Following previous works (Kim et al., 2019; Xiang et al., 2022; Sawhney et al., 2021a; Li et al., 2021), we evaluate DGDNN on three real-world datasets from two US stock markets (NASDAQ and NYSE) and one China stock market (SSE). We collect historical stock indicators from Yahoo Finance and Google Finance for all the selected stocks. We choose the stocks that span the S&P 500 and NASDAQ composite indices for the NASDAQ dataset. We select the stocks that span the Dow Jones Industrial Average, S&P 500, and NYSE composite indices for the NYSE dataset. We choose the stocks that compose the SSE 180 for the SSE dataset. The details of the three datasets are presented in Table 1.

5.2 Model Setting

Based on grid search, hyperparameters are selected using sensitivity analysis over the validation period (see Section 5.6). For NASDAQ, we set $\alpha = 2.9 \times 10^{-3}$, $\tau = 19$, $K = 9$, and $L = 8$. For NYSE, we set $\alpha = 2.7 \times 10^{-3}$, $\tau = 22$, $K = 10$, and $L = 9$. For SSE, we set $\alpha = 8.6 \times 10^{-3}$, $\tau = 14$, $K = 3$, and $L = 5$. The training epoch is set to 1200. Adam is the optimizer with a learning rate of 2×10^{-4} and a weight decay of 1.5×10^{-5}. The number of layers of Muti-Layer Perceptron is set to 3, the number of heads of Cat Attention layers is set to 3, the embedding dimension is set to 128, and full batch training is selected.

5.3 Baseline

To evaluate the performance of the proposed model, we compared DGDNN with the following baseline approaches:

Table 1: Statistics of NASDAQ, NYSE, and SSE.

	NASDAQ	NYSE	SSE
Train Period	05/2016-06/2017	05/2016-06/2017	05/2016-06/2017
Validation Period	07/2017-12/2017	07/2017-12/2017	07/2017-12/2017
Test Period	01/2018-12/2019	01/2018-12/2019	01/2018-12/2019
# Days Tr:Val:Test	252:64:489	252:64:489	299:128:503
# Stocks	1026	1737	130
# Stock Indicators	5	5	4
# Label per trading day	2	2	2

5.3.1 RNN-Based Baseline

- DA-RNN (Qin et al., 2017). A dual-stage attention-based RNN model with an encoder-decoder structure. The encoder utilizes an attention mechanism to extract the input time-series feature, and the decoder utilizes a temporal attention mechanism to capture the long-range temporal relationships among the encoded series.
- Adv-ALSTM (Feng et al., 2019a). An LSTM-based model that leverages adversarial training to improve the generalization ability of the stochasticity of price data and a temporal attention mechanism to capture the long-term dependencies in the price data.

5.3.2 Transformer-Based Baseline

- HMG-TF (Ding et al., 2021). A transformer method for modeling long-term dependencies of financial time series. The model proposes multi-scale Gaussian priors to enhance the locality, orthogonal regularization to avoid learning redundant heads in multi-head attention, and trading gap splitter to learn the hierarchical features of high-frequency data.
- DTML (Yoo et al., 2021). A multi-level context-based transformer model learns the correlations between stocks and temporal correlations in an end-to-end way.

5.3.3 GNN-Based Baseline

- HATS (Kim et al., 2019). A GNN-based model with a hierarchical graph attention mechanism. It utilizes LSTM and GRU layers to extract the temporal features as the node representation, and

[1] https://github.com/pixelhero98/DGDNN

Table 2: ACC, MCC, and F1-Score of proposed DGDNN and other baselines on next trading day stock trend classification over the test period. Bold numbers denote the best results.

Method	NASDAQ			NYSE			SSE		
	ACC(%)	MCC	F1-Score	ACC(%)	MCC	F1-Score	ACC(%)	MCC	F1-Score
DA-RNN (Qin et al., 2017)	57.59±0.36	$0.05\pm1.47\times10^{-3}$	0.56±0.01	56.97±0.13	$0.06\pm1.12\times10^{-3}$	0.57±0.02	56.19±0.23	$0.04\pm1.24\times10^{-3}$	0.52±0.02
Adv-ALSTM (Feng et al., 2019a)	51.16±0.42	$0.04\pm3.88\times10^{-3}$	0.53±0.02	53.42±0.30	$0.05\pm2.30\times10^{-3}$	0.53±0.02	52.41±0.56	$0.03\pm6.01\times10^{-3}$	0.51±0.01
HMG-TF (Ding et al., 2021)	57.18±0.17	$0.11\pm1.64\times10^{-3}$	0.59±0.01	58.49±0.12	$0.09\pm2.03\times10^{-3}$	0.59±0.02	58.88±0.20	$0.12\pm1.71\times10^{-3}$	0.59±0.01
DTML (Yoo et al., 2021)	58.27±0.79	$0.07\pm2.75\times10^{-3}$	0.58±0.01	59.17±0.25	$0.07\pm3.07\times10^{-3}$	0.60±0.01	59.25±0.38	$0.11\pm4.79\times10^{-3}$	0.59±0.02
HATS (Kim et al., 2019)	51.43±0.49	$0.01\pm5.66\times10^{-3}$	0.48±0.01	52.05±0.82	$0.02\pm7.42\times10^{-3}$	0.50±0.03	53.72±0.59	$0.02\pm3.80\times10^{-3}$	0.49±0.01
STHAN-SR (Sawhney et al., 2021a)	55.18±0.34	$0.03\pm4.11\times10^{-3}$	0.56±0.01	54.24±0.50	$0.01\pm5.73\times10^{-3}$	0.58±0.02	55.01±0.11	$0.03\pm3.09\times10^{-3}$	0.57±0.01
GraphWaveNet (Wu et al., 2019)	59.57±0.27	$0.07\pm2.12\times10^{-3}$	0.60±0.02	58.11±0.66	$0.05\pm2.21\times10^{-3}$	0.59±0.02	60.78±0.23	$0.06\pm1.93\times10^{-3}$	0.57±0.01
HyperStockGAT (Sawhney et al., 2021b)	58.23±0.68	$0.06\pm1.23\times10^{-3}$	0.59±0.02	59.34±0.19	$0.04\pm5.73\times10^{-3}$	0.61±0.02	57.36±0.10	$0.09\pm1.21\times10^{-3}$	0.58±0.02
DGDNN	**65.07±0.25**	$\mathbf{0.20\pm2.33}\times10^{-3}$	**0.63±0.01**	**66.16±0.14**	$\mathbf{0.14\pm1.67}\times10^{-3}$	**0.65±0.01**	**64.30±0.32**	$\mathbf{0.19\pm4.33}\times10^{-3}$	**0.64±0.02**

the message-passing is achieved by selectively aggregating the representation of directly adjacent nodes according to their edge type at each level.

- STHAN-SR (Sawhney et al., 2021a). A GNN-based model operated on a hypergraph with two types of hyperedges: industrial hyperedges and Wikidata corporate hyperedges. The node features are generated by temporal Hawkes attention, and weights of hyperedges are generated by hypergraph attention. The spatial hypergraph convolution achieves representation and information-spreading.
- GraphWaveNet (Wu et al., 2019). A spatial-temporal graph modeling method that captures the spatial-temporal dependencies between multiple time series by combining graph convolution with dilated casual convolution.
- HyperStockGAT (Sawhney et al., 2021b). A graph attention network utilizing the hyperbolic graph representation learning on Riemannian manifolds to predict the rankings of stocks on the next trading day based on profitability.

5.4 Evaluation Metric

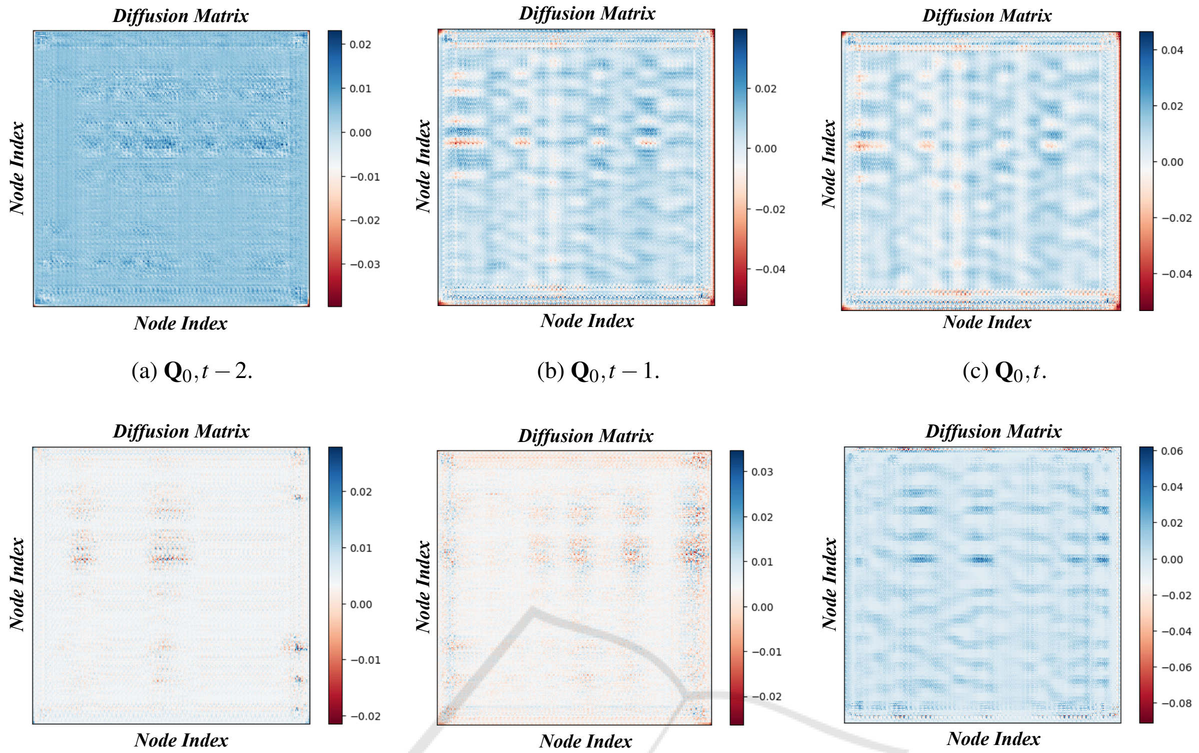
Following approaches taken in previous works (Kim et al., 2019; Xiang et al., 2022; Deng et al., 2019; Sawhney et al., 2021a; Sawhney et al., 2021b), F1-Score, Matthews Correlation Coefficient (MCC), and Classification Accuracy (ACC) are utilized to evaluate the performance of the models.

5.5 Evaluation Result

The experimental results are presented in Table 2. Our model outperforms baseline methods regarding ACC, MCC, and F1-score over three datasets. Specifically, DGDNN exhibits average improvements of 10.78% in ACC, 0.13 in MCC, and 0.10 in F1-Score compared to RNN-based baseline methods. In comparison to Transformer-based methods, DGDNN shows average improvements of 7.78% in ACC, 0.07 in MCC, and 0.05 in F1-Score. Furthermore, when contrasted with GNN-based models, DGDNN achieves average improvements of 7.16% in ACC, 0.12 in MCC, and 0.07 in F1-Score.

We can make the following observations based on experimental results. First, models such as GraphWaveNet, DTML, HMG-TF, DA-RNN, and DGDNN that obtain the interdependencies between entities during the learning process perform better in most of the metrics than those methods (HATS, STHAN-SR, HyperStockGAT, and Adv-ALSTM) with pre-defined relationships (e.g., industry and corporate edges) or without considering dependencies between entities. Second, regarding the GNN-based models, HyperStockGAT and DGDNN, which learn the graph representations in different latent spaces, perform better than those (STHAN-SR and HATS) in Euclidean space.

Fig. 3 presents visualizations of diffusion matrices across three consecutive trading days, with colors representing normalized weights. We make the following three observations. First, stocks from consecutive trading days do not necessarily exhibit similar patterns in terms of information diffusion. The distributions of edge weights change rapidly between Fig. 3a and Fig. 3b, and between Fig. 3e and Fig. 3f. Second, shallow layers tend to disseminate information across a broader neighborhood. A larger number of entries in the diffusion matrices are not zero and are distributed across the matrices in Fig 3a to Fig. 3c. In contrast, deeper layers tend to focus on specific local areas. The entries with larger absolute values are more centralized in Fig. 3d to Fig. 3f). Third, even though the initial patterns from consecutive test trading days are similar (as shown in Fig. 3b and Fig. 3c), differences in local structures result in distinctive patterns as the layers deepen (Fig. 3e and Fig. 3f), i.e., the weights of edges can show similar distributions globally, but local areas exhibit different patterns. For



(a) $\mathbf{Q}_0, t-2$.

(b) $\mathbf{Q}_0, t-1$.

(c) $\mathbf{Q}_0, t$.

(d) $\mathbf{Q}_{L-1}, t-2$.

(e) $\mathbf{Q}_{L-1}, t-1$.

(f) $\mathbf{Q}_{L-1}, t$.

Figure 3: Example normalized color maps of diffusion matrices from different layers on the NYSE dataset. $t = 03/06/2016$.

instance, in Fig. 3f, some dark blue clusters are distinguished from light blue clusters in shape and weight, which might be crucial local graph structures.

These results suggest that the complex relationships between stocks are not static but evolve rapidly over time, and the domain knowledge does not sufficiently describe the intrinsic interdependencies between multiple entities. The manually crafted fixed stock graph assumes that the stocks of the same class are connected (Livingston, 1977), neglecting the possibility that stocks change to different classes as time changes. Besides, some stocks are more critical than others in exhibiting the hierarchical nature of intra-stock dynamics (Mantegna, 1999; Sawhney et al., 2021b), which is hard to capture in Euclidean space by directly aggregating representations as the message-passing process does.

5.6 Hyperparameter Sensitivity

In this section, we explore the sensitivity of two important hyperparameters: the historical lookback window size τ and the maximum diffusion step K. These hyperparameters directly affect the model's ability to model the relations between multiple stocks. The sensitivity results of τ and K are shown in Fig. 5. Based on the sensitivity results, DGDNN consistently performs better on the three datasets when the historical lookback window size $\tau \in [14, 24]$. This coincides with the 20-day (i.e., monthly) professional financial strategies (Adam et al., 2016). Moreover, the optimal K of DGDNN varies considerably with different datasets. On the one hand, the model's performance generally improves as K grows on the NASDAQ dataset and the NYSE dataset, achieving the optimal when $K \in \{9, 10\}$. On the other hand, the model's performance on the SSE dataset reaches the peak when $K = 3$ and retains a slightly worse performance as K grows. Intuitively, the stock graph of the SSE dataset is smaller than the NASDAQ dataset and the NYSE dataset, resulting in a smaller K.

5.7 Ablation Study

The proposed DGDNN consists of three critical components: entropy-driven edge generation, generalized graph diffusion, and hierarchical decoupled representation learning. We further verify the effectiveness of each component by removing it from DGDNN. The ablation study results are shown in Fig. 4.

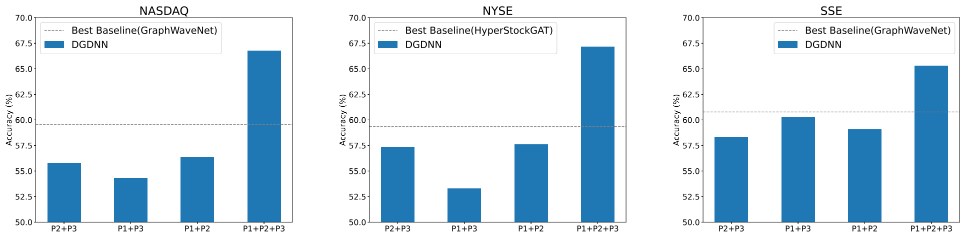


Figure 4: Results of the ablation study. Blue: P1 denotes entropy-driven edge generation, P2 denotes generalized graph diffusion, and P3 denotes hierarchical decoupled representation learning. Gray dot line: best baseline accuracy.

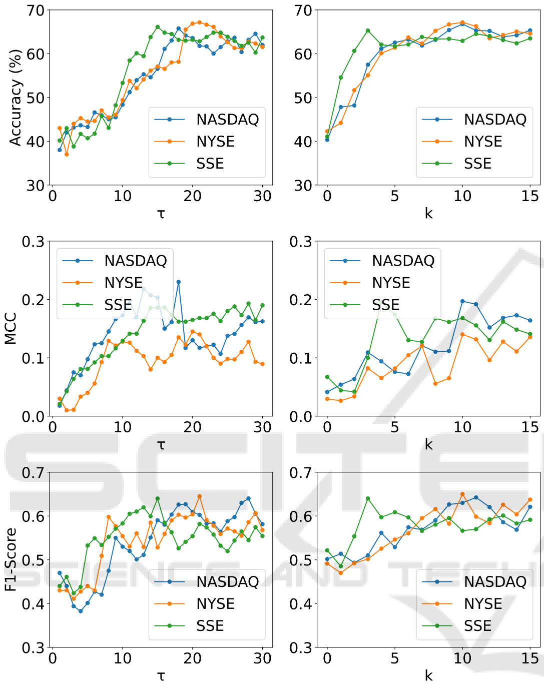


Figure 5: Sensitivity study of the historical lookback window length τ and the maximum diffusion step K over validation period.

5.7.1 Entropy-Driven Edge Generation

To demonstrate the effectiveness of constructing dynamic relations from the stock signals, we replace the entropy-driven edge generation with the commonly adopted industry-corporate stock graph using Wikidata[2] (Feng et al., 2019b). We observe that applying the industry and corporate relationships leads to an average performance reduction of classification accuracy by 9.23%, reiterating the importance of considering temporally evolving dependencies between stocks. Moreover, when testing on the NYSE dataset and the SSE dataset, the degradation of model performance is slightly smaller than on the NASDAQ dataset. According to financial studies (Jiang et al., 2011; Schwert, 2002), the NASDAQ market tends to be more unstable than the other two. This might indicate that the injection of expert knowledge works better in less noisy and more stable markets.

5.7.2 Generalized Graph Diffusion

We explore the impact of utilizing the generalized graph diffusion process. Results of the ablation study show that DGDNN performs worse without generalized graph diffusion on all datasets, with classification accuracy reduced by 10.43% on average. This indicates that the generalized graph diffusion facilitates information exchange better than immediate neighbors with invariant structures. While the performance degradation on the SSE dataset is about 38% of the performance degradation on the NASDAQ dataset and the NYSE dataset. Since the size of the stock graphs (130 stocks) of the SSE dataset is much smaller than the other two (1026 stocks and 1737 stocks), the graph diffusion process has limited improvements through utilizing larger neighborhoods.

5.7.3 Hierarchical Decoupled Representation Learning

The ablation experiments demonstrate that the model coupling the two processes deteriorates with a reduction of classification accuracy by 9.40% on the NASDAQ dataset, 8.55% on the NYSE dataset, and 5.23% on the SSE dataset. This observation empirically validates that a decoupled GNN can better capture the hierarchical characteristic of stocks. Meanwhile, this suggests that the representation transformation is not necessarily aligned with information propagation in Euclidean space. It is because different graphs exhibit various types of inter-entities patterns and intra-entities features, which do not always follow the assumption of smoothed node features (Liu et al., 2020; Xu et al., 2018; Li et al., 2018).

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

6 CONCLUSION

In this paper, we propose DGDNN, a novel graph learning approach for predicting the future trends of multiple stocks based on their historical indicators. Traditionally, stock graphs are crafted based on domain knowledge (e.g., firm-specific and industrial relations) or generated by alternative information (e.g., news and reports). To make stock graphs appropriately represent complex time-variant inter-stock relations, we dynamically generate raw stock graphs from a signal processing view considering financial theories of stock markets. Then, we propose leveraging the generalized graph diffusion process to optimize the topologies of raw stock graphs. Eventually, the decoupled representation learning scheme captures and preserves the hierarchical features of stocks, which are often overlooked in prior works. The experimental results demonstrate performance improvements of the proposed DGDNN over baseline methods. The ablation study results prove the effectiveness of each module in DGDNN. Besides financial applications, the proposed method can be easily transferred to tasks that involve multiple entities exhibiting interdependent and time-evolving features. One limitation of DGDNN is that it generates an overall dynamic relationship from multiple stock indicators without sufficiently considering the interplay between them. Notwithstanding the promising results, we plan to learn multi-relational dynamic stock graphs and allow information to be further diffused across different relational stock graphs in future work.

REFERENCES

Adam, K., Marcet, A., and Nicolini, J. P. (2016). Stock market volatility and learning. *The Journal of Finance*, 71(1):33–82.

Akita, R., Yoshihara, A., Matsubara, T., and Uehara, K. (2016). Deep learning for stock prediction using numerical and textual information. In *IEEE/ACIS 15th International Conference on Computer and Information Science*, pages 1–6.

Arora, N., Narayanan, B., and Paul, S. (2006). Financial influences and scale-free networks. In *Computational Science–ICCS 2006: Lecture Notes in Computer Science*, volume 3991, pages 16–23. Springer, Berlin, Heidelberg.

Bao, W., Yue, J., and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944.

Bo, H., Wu, Y., You, Z., McConville, R., Hong, J., and Liu, W. (2023). What will make misinformation spread: an XAI perspective. In *World Conference on Explainable Artificial Intelligence*, pages 321–337. Springer.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *34th AAAI Conference on Artificial Intelligence*, pages 3438–3445.

Chen, J., Liu, W., and Pu, J. (2022). Memory-based message passing: Decoupling the message for propagation from discrimination. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4033–4037.

Cont, R. and Bouchaud, J.-P. (2000). Herd behavior and aggregate fluctuations in financial markets. *Macroeconomic Dynamics*, 4(2):170–196.

Csiszár, I., Shields, P. C., et al. (2004). Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528.

Deng, S., Zhang, N., Zhang, W., Chen, J., Pan, J. Z., and Chen, H. (2019). Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 678–685.

Dessaint, O., Foucault, T., Frésard, L., and Matray, A. (2019). Noisy stock prices and corporate investment. *The Review of Financial Studies*, 32(7):2625–2672.

Ding, Q., Wu, S., Sun, H., Guo, J., and Guo, J. (2021). Hierarchical multi-scale gaussian transformer for stock movement prediction. In *29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4640–4646.

Feng, F., Chen, H., He, X., Ding, J., Sun, M., and Chua, T.-S. (2019a). Enhancing stock movement prediction with adversarial training. In *28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5843–5849.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y., and Chua, T.-S. (2019b). Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems*, 37(2):1–30.

Feng, S., Xu, C., Zuo, Y., Chen, G., Lin, F., and XiaHou, J. (2022). Relation-aware dynamic attributed graph attention network for stocks recommendation. *Pattern Recognition*, 121:108119.

Ferrer, R., Shahzad, S. J. H., López, R., and Jareño, F. (2018). Time and frequency dynamics of connectedness between renewable energy stocks and crude oil prices. *Energy Economics*, 76:1–20.

Gao, X., Hu, W., and Guo, Z. (2020). Exploring structure-adaptive graph learning for robust semi-supervised classification. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Huang, W., Rong, Y., Xu, T., Sun, F., and Huang, J. (2020). Tackling over-smoothing for general graph convolutional networks. *arXiv preprint arXiv:2008.09864*.

Huynh, T. T., Nguyen, M. H., Nguyen, T. T., Nguyen, P. L., Weidlich, M., Nguyen, Q. V. H., and Aberer, K. (2023). Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction.

In *16th ACM International Conference on Web Search and Data Mining*, pages 850–858.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, 106(4):620.

Jiang, B., Zhang, Z., Lin, D., Tang, J., and Luo, B. (2019). Semi-supervised learning with graph learning-convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11313–11320.

Jiang, C. X., Kim, J.-C., and Wood, R. A. (2011). A comparison of volatility and bid–ask spread for NASDAQ and NYSE after decimalization. *Applied Economics*, 43(10):1227–1239.

Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, 184:115537.

Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., and Tang, J. (2020). Graph structure learning for robust graph neural networks. In *26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 66–74.

Kim, R., So, C. H., Jeong, M., Lee, S., Kim, J., and Kang, J. (2019). HATS: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999*.

Klicpera, J., Weißenberger, S., and Günnemann, S. (2019). Diffusion improves graph learning. In *33rd International Conference on Neural Information Processing Systems (NeurIPS)*, pages 13366–13378.

Li, Q., Gama, F., Ribeiro, A., and Prorok, A. (2020a). Graph neural networks for decentralized multi-robot path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11785–11792.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *32nd AAAI Conference on Artificial Intelligence*, pages 3528–3545.

Li, Q., Tan, J., Wang, J., and Chen, H. (2020b). A multimodal event-driven LSTM model for stock prediction using online news. *IEEE Transactions on Knowledge and Data Engineering*, 33(10):3323–3337.

Li, W., Bao, R., Harimoto, K., Chen, D., Xu, J., and Su, Q. (2021). Modeling the stock relation with graph network for overnight stock movement prediction. In *29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4541–4547.

Liu, C. and Arunkumar, N. (2019). Risk prediction and evaluation of transnational transmission of financial crisis based on complex network. *Cluster Computing*, 22:4307–4313.

Liu, M., Gao, H., and Ji, S. (2020). Towards deeper graph neural networks. In *26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 338–348.

Livingston, M. (1977). Industry movements of common stocks. *The Journal of Finance*, 32(3):861–874.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1):59–82.

Mantegna, R. N. (1999). Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems*, 11:193–197.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., and Leiserson, C. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *34th AAAI Conference on Artificial Intelligence*, pages 5363–5370.

Qin, Y., Song, D., Cheng, H., Cheng, W., Jiang, G., and Cottrell, G. W. (2017). A dual-stage attention-based recurrent neural network for time series prediction. In *26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2627–2633.

Rong, Y., Huang, W., Xu, T., and Huang, J. (2019). DropEdge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*.

Roondiwala, M., Patel, H., and Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research*, 6(4):1754–1756.

Rusch, T. K., Bronstein, M. M., and Mishra, S. (2023). A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*.

Sawhney, R., Agarwal, S., Wadhwa, A., Derr, T., and Shah, R. R. (2021a). Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *35th AAAI Conference on Artificial Intelligence*, pages 497–504.

Sawhney, R., Agarwal, S., Wadhwa, A., and Shah, R. (2020). Deep attentive learning for stock movement prediction from social media text and company correlations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8415–8426.

Sawhney, R., Agarwal, S., Wadhwa, A., and Shah, R. (2021b). Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proceedings of the Web Conference*, pages 11–22.

Schwert, G. W. (2002). Stock volatility in the new millennium: how wacky is NASDAQ? *Journal of Monetary Economics*, 49(1):3–26.

Shahzad, S. J. H., Hernandez, J. A., Rehman, M. U., Al-Yahyaee, K. H., and Zakaria, M. (2018). A global network topology of stock markets: Transmitters and receivers of spillover effects. *Physica A: Statistical Mechanics and its Applications*, 492:2136–2153.

Shi, Z. and Cartlidge, J. (2022). State dependent parallel neural Hawkes process for limit order book event stream prediction and simulation. In *28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1607–1615.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., and Yu, J. (2020). Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference*, pages 1082–1092.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. In *28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1907–1913.

Xiang, S., Cheng, D., Shang, C., Zhang, Y., and Liang, Y. (2022). Temporal and heterogeneous graph neural network for financial time series prediction. In *31st ACM International Conference on Information & Knowledge Management*, pages 3584–3593.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning (ICML)*, pages 5453–5462.

Xu, Y. and Cohen, S. B. (2018). Stock movement prediction from tweets and historical prices. In *56th Annual Meeting of the Association for Computational Linguistics*, pages 1970–1979.

Ye, J., Zhao, J., Ye, K., and Xu, C. (2021). Multi-graph convolutional network for relationship-driven stock movement prediction. In *25th International Conference on Pattern Recognition*, pages 6702–6709.

Yoo, J., Soun, Y., Park, Y.-c., and Kang, U. (2021). Accurate multivariate stock movement prediction via data-axis Transformer with multi-level contexts. In *27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2037–2045.

Yue, P., Fan, Y., Batten, J. A., and Zhou, W.-X. (2020). Information transfer between stock market sectors: A comparison between the usa and china. *Entropy*, 22(2):194.

Zeng, H., Zhang, M., Xia, Y., Srivastava, A., Malevich, A., Kannan, R., Prasanna, V., Jin, L., and Chen, R. (2021). Decoupling the depth and scope of graph neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:19665–19679.

Zhang, Q., Chang, J., Meng, G., Xiang, S., and Pan, C. (2020). Spatio-temporal graph structure learning for traffic forecasting. In *34th AAAI Conference on Artificial Intelligence*, pages 1177–1185.

Zhao, J., Dong, Y., Ding, M., Kharlamov, E., and Tang, J. (2021). Adaptive diffusion in graph neural networks. *Advances in Neural Information Processing Systems*, 34:23321–23333.

Zhu, Y., Xu, W., Zhang, J., Du, Y., Zhang, J., Liu, Q., Yang, C., and Wu, S. (2021). A survey on graph structure learning: Progress and opportunities. arXiv:2103.03036 https://doi.org/10.48550/arXiv.2103.03036.