2020 IEEE International Conference on Data Mining (ICDM)

Spatiotemporal Hypergraph Convolution Network for Stock Movement Forecasting

1st Ramit Sawhney
IIIT Delhi
New Delhi, India
ramits@iiitd.ac.in

2nd Shivam Agarwal
Manipal Institute of Technology
Karnataka, India
shivam.agarwal4@learner.manipal.edu

3rd Arnav Wadhwa
MIDAS, IIIT Delhi
New Delhi, India
arnavw96@gmail.com

4th Rajiv Ratn Shah
IIIT Delhi
New Delhi, India
rajivratn@iiitd.ac.in

***Abstract*—Stock movement prediction, a widely addressed research avenue in the world of computer science and finance, it finds fundamental applications in quantitative trading and investment decision making. Predicting future trends in stock prices is a complex problem, fundamentally due to the highly stochastic and dynamic nature of the market. Advances in neural stock forecasting through deep learning models have made improvements in stock movement prediction. However, a majority of existing research treats stocks independent of each other or simplifies the complex higher-order relations between stocks in a pairwise fashion through graphs. Another limitation of recent graph-based approaches for stock movement prediction is the lack of time-aware modeling of the temporal evolution of stock prices jointly while modeling inter stock relations. To this end, we propose STHGCN: Spatio-Temporal Hypergraph Convolution Network, the first neural hypergraph model for stock trend forecasting. At the core of STHGCN, we devise a gated temporal convolution over hypergraphs for learning stock price evolution over stock relations in a time-aware manner. STHGCN significantly outperforms state-of-the-art stock forecasting methods over extensive experiments on long term real-world S&P500 index data of stocks traded in the NASDAQ and NYSE markets over 12 diverse phases. We highlight STHGCN's practical applicability through a market simulation and a latency analysis with competitive models. Furthermore, we propose a novel architecture for stock trend forecasting that can be applied across various problems in the spatiotemporal domain.**

***Index Terms*—hypergraph, stock, finance, graph neural networks, graph convolution networks, spatiotemporal convolution**

I. Introduction

The stock market, a financial ecosystem involving transactions between businesses and investors, saw a market capitalization of more than $65 trillion globally as of the year 2019.[1] The stock market presents various opportunities that increasingly attract traders and investors, who aim to utilize the potential of the market as a platform to generate profits. The development of efficient trading strategies has been an area of keen interest for an increasing number of businesses to optimize investment portfolios while maximizing profits. Stock prices have an intrinsically volatile and non-stationary nature, making their movements hard to forecast [1]. The Efficient Market Hypothesis [2] states that stock prices are indicative of all available information, suggesting the complexity of *beating the market* and accurately forecasting stock prices. However,

[1]https://data.worldbank.org/indicator/CM.MKT.LCAP.CD/

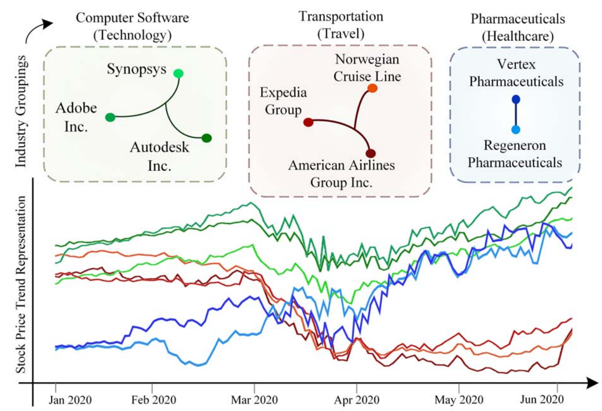

Fig. 1: Relations between stocks (nodes) belonging to the same industry (hyperedge) can be collectively represented as hypergraphs. Here, we observe how price movements of stocks belonging to the same hyperedge are correlated to each other.

deep learning demonstrates a highly promising prospect in forecasting the near-future *movement trends* of a stock [3]. Deep learning techniques have focused on leveraging latent patterns hidden in historic financial factors, analysis of online news [4], aggregation of investor sentiments, and accounting for inter-stock relationships [5]. Conventional research in stock market prediction revolves around the usage of past price data of stock to develop statistical and neural models capable of forecasting price trends [6]. Prices are driven by diverse factors that include but are not limited to company performance, historical trends, investor sentiment. Recent research shows the effectiveness of incorporating data from alternative sources such as financial news [4] and social media [7].

The majority of existing research treats stock movements to be independent of each other, contrary to true market function [8]. Often, stocks are related to each other, and there exist rich signals in relationships between stocks (or companies) [9]. For example, stocks belonging to the same industry, such as a set of travel-related stocks that observe a seasonal fall in their price, or during a worldwide pandemic, as shown in Figure 1. Studies [10], [11] demonstrate the effectiveness of modeling stock correlations for more accurate prediction.

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00057

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

Recent advances in graph-based deep learning [12] have led to the rise of graph neural networks that have successfully improved stock movement prediction by learning the latent correlations that may exist between stocks [5], [13]–[16].

Newer work [13], [14], [16] uses graphs to represent stock interactions in a pairwise fashion, simplifying the inherent higher-order relations between a group of stocks. Such a simplification of modeling inter stock (nodes) relations (edges) pairwise can also lead to densely connected graphs, potentially having a large amount of noise [5]. Stocks are often correlated through higher-order relations such as a collective group as opposed to pairwise relations. For instance, Figure 1 shows stocks belonging to a common industry collectively exhibit synchronous price movement trends. In the light of recent events regarding the COVID-19 outbreak, we observe that companies belonging to industries like travel and transportation suffer losses, leading to a downtrend in their stock prices. However, companies belonging to industries like Pharmaceuticals and Healthcare observe a rise in stock prices. Such behavior suggests the existence of *set-wise connections* among a group of stocks, for which we employ hypergraphs [17] to collectively model stocks based on industrial relations. Hypergraphs are a generalization of graphs that use hyperedges to connect multiple stocks simultaneously, as shown in Figure 1. The recently developing literature on hypergraph learning [18] presents a promising unexplored prospect for financial tasks, quantitative trading, and stock movement prediction.

Building on these gaps in existing research, we model stocks as a hypergraph, and propose STHGCN: **S**patio **T**emporal **H**yper**G**raph **C**onvolution **N**etwork, a neural framework that forecasts stock movements **(Sec. III-A)**. Building on the success of graph-based learning for stock movement prediction, STHGCN uses Spatio-temporal hypergraph convolutions **(Sec. III)** to learn the temporal evolution in stock prices and relations between stocks. We propose a novel gated temporal hypergraph convolution mechanism to model the evolution of stock movements related by a hypergraph in a time-aware manner. STHGCN learns the collective synergy between stock movements through hypergraph learning based on industrial taxonomies **(Sec. III-B)** in a time-sensitive manner **(Sec. III-D)**. Through experiments **(Sec. IV)** on real-world data **(Sec. IV-A)**, we show that STHGCN significantly outperforms state-of-the-art methods **(Sec. V)**, with an improvement of 4% in the F1 score. We demonstrate practical applicability to algorithmic trading through simulation for profit analysis **(Sec. V-C)**, and a latency speedup of approximately 28 times **(Sec. V-D)** as opposed to the state-of-the-art.

The contributions of our work can be summarized as:

- We propose a novel SpatioTemporal Hypergraph Convolution Network that models inter stock industrial relations as a hypergraph for stock movement prediction. To the best of our knowledge, STHGCN is the first hypergraph learning approach for stock movement prediction.
- We present a novel gated temporal hypergraph convolution to model related stock movements in a time-aware manner. The convolution leads to a latency reduction for stock forecasting and can be applied across other hypergraph problems involving time-evolving features.
- Through experiments on real-world S&P 500 index data in NYSE and NASDAQ stock markets, over 1,174 trading days split into 12 diverse market conditions, and we empirically highlight STHGCN's practical applicability for stock movement prediction and real-world trading.

II. Related Work

Conventional Methods in Finance: Financial models conventionally focused on technical analysis (TA) and relied only on numerical features like past prices [6], [19] and macroeconomic indicators like GDP [20]. Stock movement prediction spans various techniques [3]. These include quantitative models like *Modern Portfolio Theory* [21], *Black-Scholes model* [22], discrete [23], continuous [24], and neural approaches [6], [25]. Stock movement prediction finds practical applications including investment strategies [26], portfolio management [27]. Applications also extend beyond finance, such as forecasting presidential approval [28] and weather [29], and neuro-muscular activation modeling [30].

Contemporary Methods: Newer models based on Efficient Market Hypothesis (EMH) are categorized under fundamental analysis (FA) [31], and account for stock affecting factors beyond numerical ones such as investor sentiment through news, social media, etc. Despite their widespread success [4], [7], [32], a limitation of existing TA and FA methods is that they assume stock movements to be independent of each other. This assumption hinders the stock prediction model's ability to learn latent patterns for the study of interrelated stocks. A new line of work revolves around employing graph-based methods to represent pairwise relations between stocks using metadata, such as stock-industry information and links between company CEOs [15], [16]. These methods formulate stock movement prediction as a node classification problem and improve price-based (TA) models using graph-based deep learning approaches. For instance, [5] proposes an attention-based graph neural network for stock movement prediction. They show that all stocks are not equally correlated, and often factoring a large number of pairwise stock relations increases the noise in stock graphs, thereby reducing predictive performance. Similarly, [15] augment graph convolution networks (GCNs) with temporal convolutions and demonstrate the utility of augmenting temporal stock price evolution methods with inter-stock relations. Despite these advancements in graph-based stock movement prediction, a simplification that existing models make is that they assume stocks to be related in a pairwise fashion. The decomposition of stock data that are inherently better represented as hypergraphs in such a manner leads to a loss of vital higher-order relation information.

Hypergraph Representation and Learning: Hypergraphs have proved to be efficient in representing information where relations among data points extend beyond pairwise interactions [18], [33]–[35]. Hypergraph learning has shown immense progress in various tasks, including visual object recognition [18] and emotion recognition [36] owing to its ability to extract

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

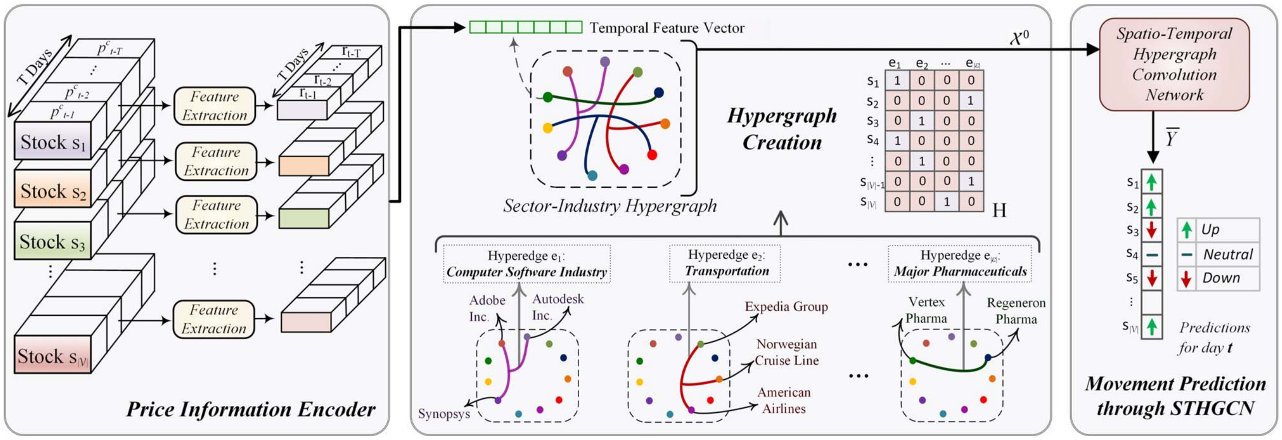

Fig. 2: An overview of the model. GRUs are added after feature extraction while using HGCN instead of STHGCN

patterns from higher-order relationships, Recent work such as [37] shows that more conventional hypergraph representation learning methods such as Deep Hyper Network Embedding [38] that model higher-order relations as a set of pairwise relations are restricted to fixed-length hyperedges leading to a lack of generalizability. A gap in existing hypergraph neural networks is that they are not specifically developed for temporally learning from time-evolving features such as daily stock prices. Limited early research such as [39] demonstrated the effectiveness of clustering stocks as hypergraphs based on stock activity data, using association rule clustering. [40] and [41] have previously represented stocks via hypergraphs by forming hyperedges that group stocks having similar price movement trends. These initial methods show the merit of representing stocks as hypergraphs, but have not explored deep learning approaches and do not use additional metadata beyond historical price signals, limiting their potential to compete with state-of-the-art methods.

III. Methodology

A. Problem Formulation

We formulate stock movement prediction as a ternary classification problem. Following [5], we use historical stock prices from an interval of T days over the range $[t-T, t-1]$ and predict the movement in stock price at the end of trading day t. Formally, for a given stock $s \in S$, the price movement from day $t-1$ to day t is defined as:

$$Y = \begin{cases} \text{up}, & r_t > l_1 \\ \text{neutral}, & l_2 < r_t \leq l_1 \\ \text{down}, & r_t \leq l_2 \end{cases} \tag{1}$$

where, $r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$ is the 1-day return of a given stock s on trading day t, indicative of the price rate change. Here, p_t is the adjusted closing price[2], it reflects the true stock price at the end of a trading day t by accounting for any corporate actions such as mergers, or stock splits [5], [25], [32]. Following [5], we define l_1 and l_2 as two threshold values to divide the stock price movements in three classes, namely: *up* corresponding to a rise in price, *down* corresponding to a fall in price, and *neutral* corresponding to a negligible change in price. The neutral class has been extensively used to account for cases where the price stays about the same [42]. We keep the same values l_1 and l_2 as [5] for empirical comparisons.

We present an overview of our proposed STHGCN model in Figure 2. In the following subsections, we first describe the hypergraph construction to model inter stock relations (**Sec.** III-B) followed by spatial hypergraph convolutions over the constructed hypergraph (**Sec.** III-C). We then show how the price features are extracted and explain the gated temporal convolution for learning the temporal evolution of stock features (**Sec.** III-D). Finally, we detail how we combine temporal and spatial hypergraph convolutions to capture temporal and spatial features for end-to-end stock movement prediction (**Sec.** III-E).

B. Stock Hypergraph Construction

Stocks that belong to the same industry tend to collectively experience similar price movement trends based on the industry's prospective performance [43]. To leverage the rich information in relations between stocks, in this work, we make use of the industries these stocks belong to and leave exploring other complex relations between stocks as one future work. Such sector-industry information can be used to define relations between stocks belonging to the same industry. Opposed to prior work [5], [15], [16], we model the relations between stocks in the same industry as a hypergraph rather than a graph. As shown in Figure 2, we create a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, where each vertex $v \in \mathcal{V}$ represents a stock $s \in S$, and each hyperedge $e \in \mathcal{E}$ represents of a subset of stocks $\{s_1, s_2, \ldots, s_n\} \in S$ that belong to the same industry. Note that each stock belongs to only one hyperedge, and the number of stocks in each hyperedge can be different. Each hyperedge e is assigned a positive weight $w(e)$ with all weights stored in a diagonal matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$. In this work, we let $\mathbf{W}$ equal the identity matrix $\mathbf{I}$ indicating

[2]https://www.investopedia.com/terms/a/adjusted_closing_price.asp

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

equal weights for all hyperedges. The hypergraph $\mathcal{G}$ can be equivalently denoted by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, with entries defined by the function $h(v,e)$ as:

$$h(v,e) = \begin{cases} 1, & v \in e \\ 0, & v \notin e \end{cases} \tag{2}$$

We find the degree of vertex v using the function $d(v)$, and store in a diagonal matrix $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. We define $d(v)$ as:

$$d(v) = \sum_{e\in\mathcal{E}} w(e)h(v,e) = \sum_{e\in\mathcal{E}} h(v,e) \tag{3}$$

We find the degree of hyperedge e via the function $\delta(e)$, and store in a diagonal matrix $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$. We define $\delta(e)$ as:

$$\delta(e) = \sum_{v\in\mathcal{V}} h(v,e) \tag{4}$$

C. Spatial Hypergraph Convolution Network (HGCN)

Hypergraphs being a generalization of usual graphs can represent higher-order relationships by connecting multiple stocks through one hyperedge [17], [44]. Thus, we model the stock price movement prediction problem as a node classification problem. This problem can be formulated as a real valued optimization problem given by [34] as:

$$\underset{f}{\operatorname{argmin}}\, \Omega(f) \tag{5}$$

where, $\Omega(f)$ is a regularizer on the hypergraph given by,

$$\Omega(f) = \frac{1}{2}\sum_{e\in\mathcal{E}}\sum_{\{u,v\in\mathcal{V}\}} \frac{w(e)h(u,e)h(v,e)}{\delta(e)}\left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2 \tag{6}$$

where, $f(\cdot)$ is a differentiable neural network [34]. Furthermore, we use the standard normalized hypergraph matrix representation $\boldsymbol{\theta}$ as:

$$\boldsymbol{\theta} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2} \tag{7}$$

The hypergraph laplacian is defined as $\boldsymbol{\Delta} = \mathbf{I} - \boldsymbol{\theta}$, $\boldsymbol{\Delta} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is a positive semi-definite quantity [17]. Therefore, $\Omega(f)$ can be normalised as [34],

$$\Omega(f) = f^T\boldsymbol{\Delta} \tag{8}$$

We use spectral decomposition $\boldsymbol{\Delta} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T$ [45] to obtain the diagonal matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2 \ldots \lambda_{|\mathcal{V}|})$ containing non-negative eigenvalues and orthonormal eigen-vectors $\boldsymbol{\Phi} = \mathrm{diag}(\phi_1, \phi_2 \ldots \phi_{|\mathcal{V}|})$.

We propose using a hypergraph convolution (HGCN) network to update a node's features by aggregating features from its neighboring nodes. Such convolutions can be considered as information exchange between related stocks as a feature propagation mechanism [46]. Thus, we apply a convolutional operator $*$ for efficient message passing between stocks [18], as shown in Figure 3 (left). From graph based spectral theory, we apply a spectral convolutional filter $\mathbf{g}$ to the input signal $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_{|\mathcal{V}|})$ in the Fourier domain given by,

$$\mathbf{g} * \mathbf{x} = \boldsymbol{\Phi}((\boldsymbol{\Phi}^T\mathbf{g}) \cdot (\boldsymbol{\Phi}^T\mathbf{x})) = \boldsymbol{\Phi}\mathbf{g}(\boldsymbol{\Lambda})\boldsymbol{\Phi}^T\mathbf{x} \tag{9}$$

where $\cdot$ denotes point-wise product and $g(\boldsymbol{\Lambda}) = \mathrm{diag}(g(\lambda_1), g(\lambda_2) \ldots g(\lambda_{|\mathcal{V}|}))$. However, in large hypergraphs, calculating eigenvalue decomposition of the hypergraph laplacian is computationally expensive [47]. To address this problem, we approximate the spectral filters in terms of truncated Chebyshev polynomials $T_k(x)$ up to the k^{th} order based on Chebyshev coefficient θ_k [18], [48]. Formally,

$$\mathbf{g} * \mathbf{x} \approx \sum_{k=0}^{K} \theta_k T_K(\bar{\boldsymbol{\Lambda}})\mathbf{x} \tag{10}$$

where, $\bar{\boldsymbol{\Lambda}} = \frac{2}{\lambda_{\max}}\boldsymbol{\Delta} - \mathbf{I}$ with $\lambda_{\max}$ denoting the largest eigenvalue of the hypergraph laplacian $\boldsymbol{\Delta}$. Furthermore, Chebyshev coefficients can be represented as a recurring polynomial given by $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0(x) = 1$ and $T_1(x) = x$. Note that we let $K = 1$ because the hypergraph Laplacian can already represent higher-order interrelations between nodes (stocks). Further, [25], [48] suggested that $\lambda_{\max} \approx 2$, as neural networks will adapt to the change in scale during training. Therefore, equation 10 simplifies to,

$$\mathbf{g} * \mathbf{x} = \theta_0\mathbf{x} + \theta_1\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{x} \tag{11}$$

where, θ_0 and θ_1 are parameters of the kernels over all nodes. In practice, it is useful to reduce the number of parameters to mitigate overfitting and reduce computational complexity [18]. Thus, we use a single parameter θ resulting in:

$$\begin{aligned} \mathbf{g} * \mathbf{x} &\approx \frac{1}{2}\theta\mathbf{D}_v^{-1/2}\mathbf{H}(\mathbf{W}+\mathbf{I})\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{x} \\ &\approx \theta\mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{x} \end{aligned} \tag{12}$$

since, we set $\mathbf{W}$ as identity matrix and $\mathbf{W} + \mathbf{I} \approx \mathbf{W}$ can be regarded as weights of hyperedges. We can generalize the above equations for any hypergraph signal $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}|\times C}$ with C-dimensional input feature vectors per node. HGCN transforms inputs features $\mathbf{X}$ to a new set of features $\mathbf{Q} \in \mathbb{R}^{|\mathcal{V}|\times F}$ where F is the number of output features per node. Formally,

$$\mathbf{Q} = \boldsymbol{\Theta} * \mathbf{X} = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}\mathbf{X}\boldsymbol{\Theta} \tag{13}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{C\times F}$ is a matrix of filter parameters which is learnt during training.

To encode temporal dependencies in stock prices over days, we use gated recurrent neural networks (GRU) [49]. We set the 1-Day day return r_τ of each stock $s \in \mathcal{S}$ as the input to the GRU and obtain output h_τ on day τ as,

$$\vec{h}_\tau = \overrightarrow{GRU}(r_\tau, \vec{h}_{\tau-1}) \quad t - T \leq \tau \leq t - 1 \tag{14}$$

We use the last state h_T of the above GRU as a feature vector for each stock. These temporal features are used as inputs to the above hypergraph convolution mechanism. However, the inherent sequential nature of RNN-based models precludes parallelization within the sequence, leading to a higher latency [50]. Building on the rising interest in low latency stock trading, it is imperative to consider the latency involved in

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

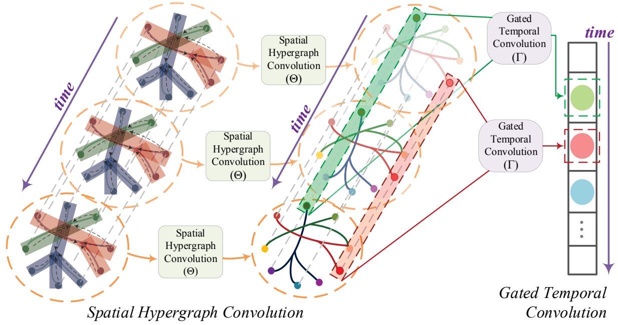

Fig. 3: Spatial (left) and gated temporal hypergraph convolution (right) for extracting spatiotemporal dependencies.

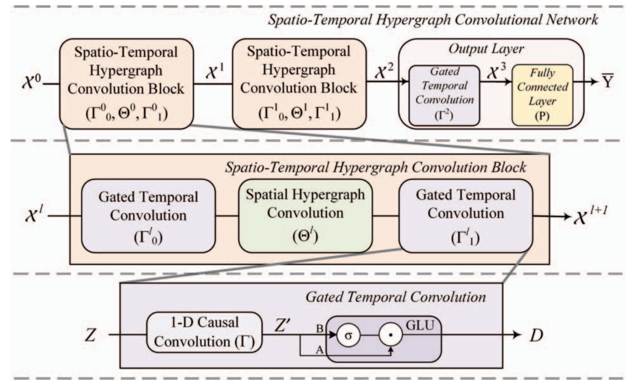

Fig. 4: Spatio-Temporal Hypergraph Convolution Block

stock prediction [51].[3] Thus, we propose a gated temporal convolution over the hypergraph to encode sequential dependencies in historic prices. We describe this next.

D. Gated Temporal Convolution

In contrast to RNNs, CNNS do not depend on the computations of the previous time-step and therefore allow parallelization over every element in the sequence [52]. To this end, we propose a gated temporal convolution, which is shown in Figure 3 (right). It consists of a 1-dimensional *causal* convolution with kernel width K_t to model the temporal dependencies in price return features. We use causal convolutions to ensure that the layer does not select features from future time-steps to prevent information "leakage" from future features to the past [53], [54]. The 1D causal convolution explores K_t neighboring features of each time-step in the input sequence. Note that this operation is done without padding, which shortens the sequence length by $K_t - 1$. Studies have found that past stock prices on each day have a different impact on future stock trends [55]. To this end, we use a gated linear unit (GLU) non-linearity after the 1D causal convolution layer. The GLU consists of gating mechanisms that highlight important information in the price feature vectors [56]. We summarize the architecture of the gated temporal convolution in Figure 4 (bottom). The input to the gated temporal convolution layer is a sequence of features $\mathbf{Z} \in \mathbb{R}^{T \times C}$, where T is the sequence length, and C is the number of features per time-step. We design a trainable convolution kernel $\mathbf{\Gamma} \in \mathbb{R}^{K_t \times C \times 2F}$ that transforms the input $\mathbf{Z}$ to output $\mathbf{Z}' \in \mathbb{R}^{(T-K_t+1) \times 2F}$ where, $2F$ is the number of output features per time-step. As shown in Figure 4 (bottom), the output $\mathbf{Z}'$ is split into two parts: $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{(T-K_t+1) \times F}$ with the same number of features in each, and act as inputs to the GLU. The GLU produces the output $\mathbf{D} \in \mathbb{R}^{(T-K_t+1) \times F}$ given by:

$$\mathbf{D} = \mathbf{\Gamma} *_{\boldsymbol{\tau}} \mathbf{Z} = \mathbf{A} \cdot \sigma(\mathbf{B}) \tag{15}$$

[3] Low latency trading refers to making decisions within milliseconds, where faster inference can lead to quicker reactions to market opportunities. Sources state that every millisecond lost results in \$100m per annum in lost opportunity. Details: https://en.wikipedia.org/wiki/Low_latency_(capital_markets)

where, $*_{\boldsymbol{\tau}}$ denotes gated temporal convolution and, $\cdot$ represents point-wise product. We use the logistic sigmoid $\sigma(\mathbf{B}) = \frac{1}{1+e^{-\mathbf{B}}}$ as a gating mechanism to control features in $\mathbf{A}$ that correspond to more important time-steps in the input $\mathbf{Z}'$ for stock movement prediction. Furthermore, several convolution layers can be stacked to increase the effective receptive field of the model over larger time steps [57].

We use the above gated temporal convolution layer for all stocks by constructing a 3-D matrix $\mathcal{Z} \in \mathbb{R}^{T \times |\mathcal{V}| \times C}$ that represents temporal features of all stocks. We generalise the gated temporal convolution to 3-D input matrix $\mathcal{Z}$ by using same kernel $\mathbf{\Gamma}$ for features of every stock in parallel producing the output $\mathcal{D} \in \mathbb{R}^{(T-K_t+1) \times |\mathcal{V}| \times F}$ given as, $\mathcal{D} = \mathbf{\Gamma} *_{\boldsymbol{\tau}} \mathcal{Z}$. We use calligraphic symbols (Eg. $\mathcal{D}$) for 3-D matrices.

E. Spatio Temporal Hypergraph Convolution (STHGCN)

We combine gated temporal and spatial hypergraph convolutions as a spatio-temporal hypergraph convolution (STH-Conv) block to jointly learn the correlations between related stocks and the temporal evolution of stock prices. We first describe a single STH-Conv block that is stacked to build STHGCN. As shown in Figure 4, the STH-Conv block consists of a hypergraph convolution (Sec III-C), capturing stock correlations and a convolution along the time axis (Sec III-D) exploits sequential dependencies from recent days. The spatial hypergraph convolution is used to couple the two temporal convolutions to achieve faster propagation of spatiotemporal features. Additionally, layer normalization is implemented within each STH-Conv block to mitigate overfitting. Formally, the l^{th} STH-Conv block transforms a 3-D matrix of features of all stocks $\mathcal{X}^l \in \mathbb{R}^{T \times |\mathcal{V}| \times C^l}$ to an output $\mathcal{X}^{l+1} \in \mathbb{R}^{(T-2(K_t-1)) \times |\mathcal{V}| \times C^{l+1}}$ through temporal and spatial hypergraph convolutions. Here, C^l and C^{l+1} are the number of input and output features per time-step of block l, respectively. We define the transformation made by the STH-Conv block as:

$$\mathcal{X}^{l+1} = \mathbf{\Gamma}^l_1 *_{\boldsymbol{\tau}} \mathrm{ReLU}(\Psi^l) \tag{16}$$

where, Ψ^l is given as:

$$\Psi^l = \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2}(\mathbf{\Gamma}^l_0 *_{\boldsymbol{\tau}} \mathcal{X}^l)\mathbf{\Theta}^l \tag{17}$$

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

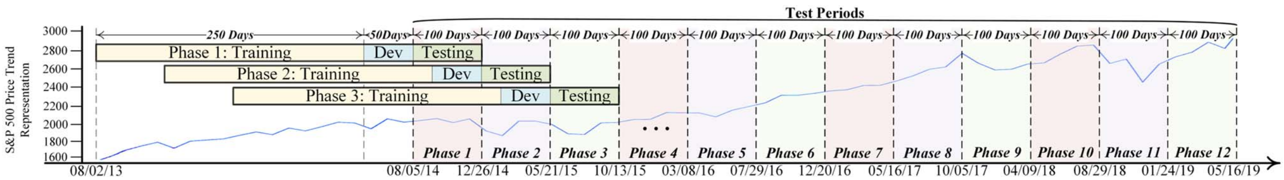

Fig. 5: Visualization of the dataset arrangement for all experiments covering 12 diverse phases of market activity. The blue line indicates the varying daily close price for the S&P 500 index.

TABLE I: Statistics about the Hypergraph

Hyperedge Property	***Value***
Number	110
Max. Degree	28
Mean Degree	3.845

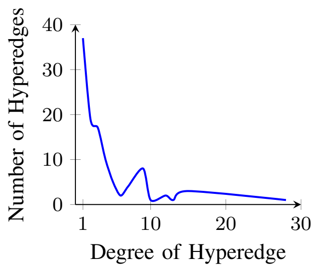

Fig. 6: Hyperedge degree distribution

where, $\mathbf{\Gamma}_0^l, \mathbf{\Gamma}_1^l$ are temporal convolution kernels within the block l. ReLU$(\cdot)$ denotes rectified linear unit activation [58]. To apply the spatial hypergraph convolution to the temporal input features $\mathcal{X}^l$, we impose the same convolution parameter $\mathbf{\Theta}$ to all time-steps of the input features in parallel as shown in Figure 3 (right). We stack two such STH-Conv blocks, followed by an output layer. The output layer consists of a gated temporal convolution and a single layer feed-forward neural network, which projects the output of the last STH-Conv block to a single-step prediction. The STHGCN is fed a 3-D input feature matrix $\mathcal{X}^0 \in \mathbb{R}^{T\times|\mathcal{V}|\times 1}$; constructed using daily price returns r_τ and predicts stock price movements $\overline{\boldsymbol{Y}} \in \mathbb{R}^{|\mathcal{V}|\times 3}$ as *[up, down, neutral]*:

$$\overline{Y} = \text{Softmax}((\mathbf{\Gamma}^2 *_{\boldsymbol{\tau}} \mathcal{X}^2)\mathbf{P}^T + b) \tag{18}$$

where, $\mathbf{P}$ and b are learnable weights and bias, respectively of a single layer feed forward neural network. $\mathbf{\Gamma}^2$ is the temporal convolution kernel that projects the output of the last STH-Conv block to a single-step prediction. $\mathcal{X}^2$ is the output produced by the last STH-Conv block. We use cross-entropy loss for our node (stock) classification task, defined as:

$$L_{cse} = -\sum_{i=1}^{|\mathcal{V}|}\sum_{j=1}^{3} Y_{ij} ln(\overline{Y}_{ij}) \tag{19}$$

where, Y_{ij} represents the true stock price movement label and $|\mathcal{V}|$ is the total number of stocks in the hypergraph.

IV. Experiments

A. Dataset

Price data: We use the dataset created by HATS [5] for training and evaluating HGCN and STHGCN. The dataset contains 431 stocks from the S&P 500 companies. We sample historical price movements from 08/02/2013 to 05/16/2019 producing 1,174 trading days using Yahoo finance[4].

Following [5], we divide the entire 1,174 trading days into 12 smaller phases spanning diverse market conditions, as shown in Figure 5. We divide the dataset into 12 smaller overlapping datasets by shifting a 350 days sliding window over the entire trading period [5]. In each 350 day window, thus created the first 250 days correspond to training samples, the next 50 for development, and the last 100 for evaluation. For all models, we use a lookback period T of 50 days to construct price feature vectors.

Sector-Industry data: To cluster stocks and represent them via hyperedges of hypergraphs, we make use of the sector industry grouping data from MSCI - Global Industry Classification Standard[5] (GICS) listings. The GICS industry taxonomy is a four-tiered hierarchical industry classification system. It is adopted by the S&P 500 to classify companies based on their principal business activities. Here, companies are classified among 11 sectors, 24 industry groups, 69 industries, and 158 sub-industries. For the 431 stocks we study, we identify 110 possible sector-industry groupings into which the stocks can be clustered. We summarise the statistics about the obtained hypergraph in Table I. We show the distribution of hyperedge degree across all stocks in Figure 6

B. Training Setup

All experiments are performed on a Tesla P100 GPU. We apply grid search to find the best hyperparameters for all models based on the F1 score obtained on the validation split for each phase. Following [5], the length of the price feature is 50 based on the lookback of $T = 50$ days. All models are trained and evaluated phase-wise over the 12 different phases. We use Adam optimizer [59] to optimize the cross-entropy loss in an end-to-end fashion for all models.

Settings for HGCN (Sec. III-C): HGCN uses independent GRUs for each stock. The hidden size of all GRUs is set to **64**. The Adam optimiser is set with default values $\beta_1 = \mathbf{0.9}$, $\beta_2 = \mathbf{0.999}$, $\epsilon = \mathbf{1e-8}$, weight decay $= \mathbf{5e-4}$ and learning rate of $\mathbf{2e-4}$.

Settings for STHGCN (Sec. III-E): We train both the STHGCN blocks (Figure 4 with the same configuration. The size of all convolution kernels is set to **3**. The output space of both the gated temporal convolutions and that of spatial

[4]https://finance.yahoo.com/industries
[5]https://www.msci.com/gics

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

hypergraph convolution has a dimension of $\mathbf{64}$. The last fully-connected layer's output size is set to $\mathbf{3}$ to categorize the trend into 3 classes. All convolution kernels are initialised from $\mathcal{U}(-\sqrt{k}, \sqrt{k})$ where $k = \frac{1}{kernel\_size}$ and $\mathcal{U}$ denotes uniform distribution. We use Adam optimizer with the same values of HGCN (described above), with the learning rate set to $\mathbf{2e-4}$.

C. Evaluation Metrics

Following previous work [5], [18], we compare against the F1 score for stock movement classification. We also compare against the Sharpe Ratio to assess practical applicability in real-world markets through a market simulation. We assess model profitability on the test data of the S&P 500 index. Following [5], [25], we create a neutralized portfolio based on the predictions obtained by the model. Then, 15 stocks with the highest *up* class probability are selected, and a long position is taken. In other words, we will buy one share of the 15 stocks at the closing price of the current trading session and sell it on the next day's closing price. To neutralize the portfolio, 15 companies with the highest down class probability are chosen, and a short sell is performed[6]. A short sell is a transaction in which the strategy sells a borrowed stock in the current trading session in anticipation of a price decline; and then buys the stock in the next day's trading session. The Sharpe ratio is a measure of the return of a portfolio compared to its risk [60]. We calculate the Sharpe ratio by computing the earned return R_a in excess of a risk-free return[7] R_f as:

$$\text{Sharpe Ratio}_a = \frac{E[R_a - R_f]}{std[R_a - R_f]} \tag{20}$$

In calculating the Sharpe ratio during the market simulation, we follow some commonly used assumptions [5], [15]. First, we assume that the market is always sufficiently liquid such that the stock bought (shorted) at the closing price of day $t-1$ will be sold at the closing price of day t. Second, we ignore transaction costs, since, for US stocks, the trading fees through brokers are relatively low (for example, Interactive Brokers charge only $\$0.005$ per share).[8] Third, we assume that only 1 share will be traded and that the trader will always have sufficient money for the transaction.

D. Baselines

Price only models:

- **MLP [61]:** A simple multi layer perceptron consisting of two hidden layers of 8 and 16 hidden neurons respectively, followed by one prediction layer.
- **CNN [62]:** A convolution neural network is fed historic stock return as input. We use a 4 layered CNN; 2 convolutions with a filter size of 32 and 8, respectively, and 2 pooling operations. Number of kernels is 5 throughout.
- **LSTM [63]:** A two layered long short term memory network with hidden size 128 that is fed the past historic return for the past 50 days sequentially.

Graph and Hypergraph based models:

- **GCN [13]:** A graph convolution network to learn from the relationships between the stocks. The historic return for the past 50 days for each stock is set as the node feature for node classification by the GCN.
- **GCN20 [5]:** A variant of the above defined GCN model. The best 20 performing relationship types from all combinations are selected based on validation F1 score.
- **TGC [15]:** A graph neural network that incorporates temporal graph convolutions to model company relations based on the current state of the relation between stocks over the time evolving stock prices.
- **HATS [5]:** State-of-the-art hierarchical graph attention model that uses a multi-graph to represent different kinds of relations. It uses hierarchical attention to selectively aggregate information from different relationship types.
- **HGCluster [40]:** HGCluster creates hyperedges between stocks exhibiting the same movement trend *(up, down, neutral)* for a fixed lookback period of $T = 50$ days. HGCluster frames stock price prediction as a hypergraph clustering algorithm, where clusters of stocks are generated based on historic stock movement. A random seed is selected in each cluster and the predictions for all stocks in that cluster are the same as that of the random seed.

V. Results and Analysis

A. Performance Comparison with Baselines

Table II shows the performance of the proposed HGCN and STHGCN models with the baselines on test sets across 12 phases averaged over 10 different runs. We observe that, on average graph-based methods outperform price only models (MLP, CNN, and LSTM), empirically validating the effectiveness of factoring in inter stock dependence. We also note that augmenting price-based stock movement predictions with graph-based learning does not always lead to superior results. On average, the GCN and TGC models perform at par with the price-only baselines. Our observations are in line with those reported in [5], that considering all relations in a pairwise fashion may lead to densely connected networks adding noise. The GCN20 pruned graph variant that has fewer edges validates the presence of noise in densely connected stock networks as it outperforms both the unpruned GCN and TGC variants. Similarly, HATS, the state-of-the-art method, selectively aggregates information from different relations using attention mechanisms, verifying that graphs that model stock relations in a pairwise fashion might contain noise.

We observe that both HGCN and STHGCN outperform all the baselines, highlighting the effectiveness of representing industry-based stock relations as hypergraphs rather than graphs. We postulate this to HGCN's ability to collectively model industry-based stock relations more realistically [64] as hypergraphs require 1 hyperedge compared to $\binom{n}{2}$ edges required by graphs to represent n stocks in an industry, leading to less turbulent information propagation in graphs [65]. This empirical observation justifies the hypothesis of using hypergraphs to model stock inter-relations. Furthermore, F1 score

[6]Short sell: https://en.wikipedia.org/wiki/Short_(finance)

[7]We use treasury bill rates, obtained from https://www.treasury.gov

[8]https://www.stockbrokers.com/guides/free-stock-trading

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Mean F1 scores across 12 phases over 10 runs. **Bold** and *Italics* denote the best and second best results respectively.

Phase	MLP [61]	HGCluster [40]	CNN [62]	LSTM [25]	GCN [13]	GCN20 [5]	TGC [15]	HATS [5]	HGCN	STHGCN
1	0.288	0.220	0.311	0.317	0.287	0.316	0.311	*0.331*	0.315	**0.330**
2	0.286	0.271	0.321	0.323	0.307	0.334	0.309	*0.335*	0.333	**0.342**
3	0.276	0.277	0.294	0.306	0.269	0.311	0.224	0.310	*0.329*	**0.338**
4	0.281	0.279	0.318	0.303	0.294	0.324	0.297	0.327	*0.337*	**0.342**
5	0.287	0.290	0.335	0.333	0.312	0.345	0.333	*0.350*	0.335	**0.366**
6	0.286	0.227	0.327	0.323	0.291	0.314	0.280	*0.339*	0.330	**0.355**
7	0.288	0.286	0.311	0.317	0.287	0.316	0.311	0.331	*0.368*	**0.381**
8	0.286	0.284	0.321	0.323	0.307	0.334	0.309	0.335	*0.337*	**0.355**
9	0.274	0.276	0.239	0.279	0.298	0.316	0.285	*0.322*	0.309	**0.335**
10	0.253	0.255	0.213	0.313	0.300	0.327	0.295	0.324	*0.333*	**0.339**
11	0.250	0.251	0.227	0.300	0.271	0.303	0.258	0.309	**0.329**	*0.327*
12	0.268	0.270	0.292	0.297	0.296	0.327	0.327	*0.340*	0.332	**0.343**
Average	0.277	0.265	0.292	0.311	0.293	0.322	0.295	0.329	*0.332*	**0.346**

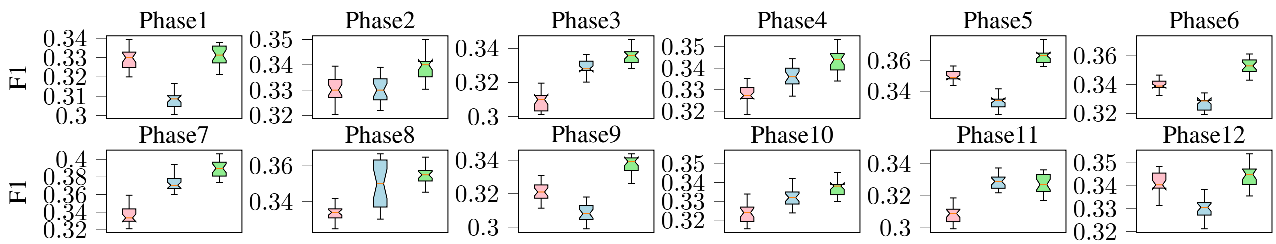

Fig. 7: Distribution of F1 scores with confidence intervals of the state-of-the-art model HATS (left, red), our proposed models HGCN (middle, blue) and STHGCN (right, green) over 10 different runs. Notched box plots best viewed in color.

improvements of the STHGCN over all baselines and HGCN display the utility of the proposed gated temporal convolution to capture the time-evolving stochastic stock prices.

B. Phase wise Comparison

Figure 7 illustrates the F1 scores obtained by HATS, HGCN, and STHGCN, along with their confidence intervals over ten different runs for a comprehensive evaluation across the 12 different phases. We see that HGCN significantly ($p < 0.05$) outperforms HATS on multiple phases (3, 4, 7, 8, 10, and 11). Through these comparisons between the HGCN and HATS, we observe that hypergraphs do show promise in modeling stock relations for stock movement prediction under most market conditions. However, based on the results, it is not easy to conclusively ascertain if HGCN outperforms HATS. We observe a statistically significant ($p < 0.05$) improvement in the F1 score as compared to HATS across all phases except phases 1. STHGCN outperforms both HATS and HGCN over multiple runs and diverse market conditions (Figure 5), empirically validating STHGCN's ability to jointly learn from stock hypergraphs and the temporal price evolution using the proposed spatiotemporal hypergraph convolutions. These improvements demonstrate the utility of the proposed gated temporal convolution for hypergraph learning problems.

C. Profitability

We examine STHGCN's practical applicability to real-world trading under 12 diverse market conditions through simulation, as described in Section IV-C. Table III shows that on average, STHGCN achieves the highest Sharpe ratio compared to HATS as well as GCN20, and performs the best in most phases. The overall improvements in profit reiterate the effectiveness of

TABLE III: Annualized Sharpe ratio (risk-adjusted return) across all 12 phases. **Bold** indicates the best results. The green gradient represents a higher profit, and red indicates a loss.

Phase	LSTM	GCN	GCN20	TGC	HATS	STHGCN
1	2.355	−0.280	−0.101	−0.503	**2.479**	*2.475*
2	4.065	2.47	**4.900**	3.052	*4.390*	3.031
3	1.064	−0.247	0.299	−0.179	**1.250**	*1.190*
4	2.201	0.028	0.317	0.108	*2.396*	**2.436**
5	−0.445	−0.709	*0.622*	−0.413	0.408	**1.650**
6	1.196	1.832	*2.039*	**2.943**	1.694	1.967
7	0.835	0.407	−0.910	−0.661	*2.033*	**2.221**
8	2.197	1.374	*3.187*	**4.131**	1.483	1.269
9	0.491	0.776	−0.689	−0.361	*0.806*	**1.745**
10	−0.567	0.326	1.357	−1.202	*1.438*	**2.036**
11	−0.198	2.578	1.305	*3.337*	**3.614**	2.761
12	0.633	0.806	−0.820	1.779	*1.901*	**1.967**
Average	1.152	0.780	0.958	1.002	*1.991*	**2.065**

gated temporal convolutions and representing stock relations as hypergraphs, similar to the gains obtained for F1 scores. Interestingly, both HATS and STHGCN do not realize a loss under any of the 12 diverse market phases. We also observe that STHGCN achieves larger profits (phases 9 and 10) but also underperforms under certain market conditions (phases 1, 2, and 8). We postulate that this might be due to the temporal variations in stock relations. Under specific market phases, the hypergraph representation of stocks may be more accurate and characteristic of the collective macro impact on stocks, say under a global pandemic, for example (Figure 1). This dynamic nature of stock relations highlights a limitation of STHGCN, where the relationships are static in nature despite the temporal evolution of prices. Dynamically evolving representations of stock relations present future research direction to steer profitability across ever-changing market conditions.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

TABLE IV: Running time comparison of proposed models with the state-of-the-art.

Model	Train time↓ (secs/epoch)	Test time↓ (secs)
HGCN	250	0.45
HATS	70	0.16
STHGCN	**2.5**	**0.002**

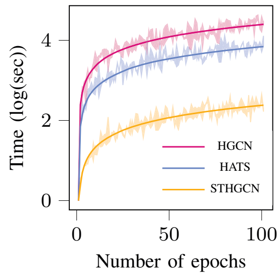

Fig. 8: Training time comparisons over epochs

D. Latency Analysis: Probing Model Efficiency

In the real world, trading and stock movement forecasting rely not only on the predictive power of models but also on their latency. The latency of a model impacts the profit-making capability of models as their speed modulates stock movement prediction models' ability to react faster to market events [51]. To this end, we summarise the training and inference time of HATS, HGCN, and STHGCN in Table IV. We observe that among HATS and HGCN, HATS is more efficient in terms of run-time as it uses a shared GRU across all stocks. We observe that STHGCN is considerably faster than both HATS (28x) and HGCN (100x). This significant improvement is because gated temporal convolutions can achieve fully parallel training rather than depending on chain structures in RNN [52]. Furthermore, Figure 8 shows that STHGCN is consistently faster than HATS and HGCN as we increase the number of epochs. The time efficiency of the proposed gated temporal convolution builds the case for STHGCN to be used in practice for algorithmic high-frequency trading that is latency-sensitive. The substantially reduced inference time sets a precedent for future work for designing trading strategies in more realistic high-frequency intraday settings.

VI. Conclusion and Future Work

We study stock movement prediction by modeling stocks via *hypergraphs* based on industrial relations. We propose STHGCN, the first neural architecture for stock forecasting that utilizes spatiotemporal hypergraph convolutions to model the temporal evolution of stock prices while accounting for relationships between connected stocks. We devise a novel gated temporal convolution component over hypergraphs that can be generalized for spatiotemporal feature learning over hypergraphs across problems in varying domains. We present quantitative comparisons with numerous baselines on real-world S&P 500 data, ranging over 1,174 trading days. We empirically demonstrate that STHGCN outperforms the state-of-the-art methods via 4% improvement in prediction performance based on the F1 score. Based on the improvements in performance, increased profit generation capability, and model latency reduction (28x), we set forth the practical applicability of STHGCN as a tool for algorithmic trading.

We further aim to explore relationships between stocks beyond industrial relations and account for other modalities affecting stocks like online financial news and social media data. Another interesting line of future work would be to explore hypergraph pooling operations for forecasting the prices of stock indexes, which are essentially a collection of multiple stocks. Given STHGCN's generalizability over hypergraph networks, we would like to explore the performance of STHGCN in other tasks such as session-based recommender systems, traffic prediction, online user-item profiling, and time-evolving citation networks in the future.

VII. Acknowledgement

Rajiv Ratn Shah is partly supported by the Infosys Center for AI and the Center of Design and New Media at IIIT Delhi.

References

[1] K. Adam, A. Marcet, and J. P. Nicolini, "Stock market volatility and learning," *The Journal of Finance*, vol. 71, no. 1, pp. 33–82, Jan. 2016. [Online]. Available: https://doi.org/10.1111/jofi.12364

[2] B. G. Malkiel, "Efficient market hypothesis," in *Finance*. Springer, 1989, pp. 127–134.

[3] W. Jiang, "Applications of deep learning in stock market prediction: recent progress," *ArXiv*, vol. abs/2003.01859, 2020.

[4] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu, "Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 261–269.

[5] R. Kim, C. H. So, M. Jeong, S. Lee, J. Kim, and J. Kang, "Hats: A hierarchical graph attention network for stock movement prediction," *ArXiv*, vol. abs/1908.07999, 2019.

[6] T.-T. Nguyen and S. Yoon, "A novel approach to short-term stock price movement prediction using transfer learning," *Applied Sciences*, vol. 9, no. 22, p. 4745, Nov 2019.

[7] 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 (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1970–1979.

[8] F. Diebold and K. Yilmaz, "On the network topology of variance decompositions: Measuring the connectedness of financial firms," *Journal of Econometrics*, vol. 182, no. 1, pp. 119–134, 2014, cited By 416.

[9] A. Nobi, S. E. Maeng, G. G. Ha, and J. W. Lee, "Effects of global financial crisis on network structure in a local stock market," *Physica A: Statistical Mechanics and its Applications*, vol. 407, pp. 135 – 143, 2014.

[10] J. D. Noh, "Model for correlations in stock markets," *Phys. Rev. E*, vol. 61, pp. 5981–5982, May 2000.

[11] J. Kwapień, S. Drożdż, and P. Oświe¸cimka, "The bulk of the stock market correlation matrix is not pure noise," *Physica A: Statistical Mechanics and its Applications*, vol. 359, p. 589–606, Jan 2006. [Online]. Available: http://dx.doi.org/10.1016/j.physa.2005.05.090

[12] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[13] 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*. ACM, oct 2018.

[14] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 5165–5175.

[15] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Transactions on Information Systems*, vol. 37, no. 2, p. 1–30, Mar 2019.

[16] D. Matsunaga, T. Suzumura, and T. Takahashi, "Exploring graph neural networks for stock market predictions with rolling window analysis," *ArXiv*, vol. abs/1909.10660, 2019.

[17] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*. ACM Press, 2006.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.

[18] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, p. 3558–3565, Jul 2019.
[19] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of lstm," *International Journal of Machine Learning and Cybernetics*, 11 2019.
[20] E. Hoseinzade, S. Haratizadeh, and A. Khoeini, "U-cnnpred: A universal cnn-based predictor for stock markets," *ArXiv*, vol. abs/1911.12540, 2019.
[21] E. J. Elton, M. J. Gruber, S. J. Brown, and W. N. Goetzmann, *Modern portfolio theory and investment analysis*. John Wiley & Sons, 2009.
[22] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of political economy*, vol. 81, no. 3, 1973.
[23] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.
[24] L. B. Andersen, "Efficient simulation of the heston stochastic volatility model," *Available at SSRN 946405*, 2007.
[25] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, Oct. 2018. [Online]. Available: https://doi.org/10.1016/j.ejor.2017.11.054
[26] D. C. Blitz and P. Van Vliet, "The volatility effect," *The Journal of Portfolio Management*, vol. 34, no. 1, pp. 102–113, 2007.
[27] A. Hocquard, S. Ng, and N. Papageorgiou, "A constant-volatility framework for managing tail risk," *The Journal of Portfolio Management*, 2013.
[28] P. Gronke and J. Brehm, "History, heterogeneity, and presidential approval: a modified arch approach," *Electoral Studies*, 2002.
[29] J. W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Transactions on Power systems*, vol. 17, no. 3, pp. 626–632, 2002.
[30] G. Elliott and A. Timmermann, *Handbook of economic forecasting*. Elsevier, 2013.
[31] I. D. Dichev and V. W. Tang, "Earnings volatility and earnings predictability," *Journal of accounting and Economics*, vol. 47, no. 1-2, pp. 160–181, 2009.
[32] Y. Qin and Y. Yang, "What you say and how you say it matters: Predicting financial risk using verbal and vocal cues," in *57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, 2019, p. 390.
[33] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 1511–1522.
[34] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 1601–1608.
[35] S. Bai, F. Zhang, and P. Torr, "Hypergraph convolution and hypergraph attention," *ArXiv*, vol. abs/1901.08150, 2019.
[36] J. Shao, J. Zhu, Y. Wei, Y. Feng, and X. Zhao, "Emotion recognition by edge-weighted hypergraph neural network," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 2144–2148.
[37] R. Zhang, Y. Zou, and J. Ma, "Hyper-sagnn: a self-attention based graph neural network for hypergraphs," in *International Conference on Learning Representations*, 2020.
[38] K. Tu, P. Cui, X. Wang, F. Wang, and W. Zhu, "Structural deep embedding for hyper-networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
[39] E. Han, G. Karypis, V. Kumar, and B. Mobasher, "Clustering based on association rule hypergraphs," in *DMKD*, 1997.
[40] Y. Luo, J. Hu, X. Wei, D. Fang, and H. Shao, "Stock trends prediction based on hypergraph modeling clustering algorithm," in *2014 IEEE International Conference on Progress in Informatics and Computing*, 2014, pp. 27–31.
[41] Yang Shen, J. Hu, Yanan Lu, and Xiaofeng Wang, "Stock trends prediction by hypergraph modeling," in *2012 IEEE International Conference on Computer Science and Automation Engineering*, 2012, pp. 104–107.
[42] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: A review and taxonomy of prediction techniques," *International Journal of Financial Studies*, vol. 7, no. 2, p. 26, May 2019.
[43] M. Livingston, "Industry movements of common stocks," *The Journal of Finance*, vol. 32, no. 3, pp. 861–874, 1977.
[44] C. Zhang, S. Hu, Z. G. Tang, and T.-H. H. Chan, "Re-revisiting learning on hypergraphs: Confidence interval and subgradient method," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017.
[45] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. USA: Johns Hopkins University Press, 1996.
[46] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 1263–1272.
[47] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 3844–3852.
[48] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ArXiv*, vol. abs/1609.02907, 2017.
[49] C. L. Giles, S. Lawrence, and A. C. Tsoi, *Machine Learning*, vol. 44, no. 1/2, pp. 161–183, 2001.
[50] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *ICML*, 2017, pp. 933–941.
[51] J. Hasbrouck and G. Saar, "Low-latency trading," *Journal of Financial Markets*, vol. 16, no. 4, pp. 646 – 679, 2013, high-Frequency Trading.
[52] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 1243–1252.
[53] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2016.
[54] D. Daiya, M.-S. Wu, and C. Lin, "Stock movement prediction that integrates heterogeneous data sources using dilated causal convolution networks with attention," in *ICASSP 2020*, 2020.
[55] Z. Zhao, R. Rao, S. Tu, and J. Shi, "Time-weighted lstm model with redefined labeling for stock trend prediction," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2017, pp. 1210–1217.
[56] X. Zhang and J. You, "A gated dilated causal convolution based encoder-decoder for network traffic forecasting," *IEEE Access*, vol. 8, pp. 6087–6097, 2020.
[57] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *ArXiv*, vol. abs/1803.01271, 2018.
[58] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: JMLR Workshop and Conference Proceedings, 11–13 Apr 2011, pp. 315–323.
[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
[60] W. F. Sharpe, "The sharpe ratio," *Journal of portfolio management*, vol. 21, no. 1, pp. 49–58, 1994.
[61] K. Schierholt and C. H. Dagli, "Stock market prediction using different neural network classification architectures," in *IEEE/IAFE 1996 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, 1996, pp. 72–78.
[62] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*. IEEE, 2017, pp. 1643–1647.
[63] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
[64] E. Bautu, S. Kim, A. Bautu, H. Luchian, and B. Zhang, "Evolving hypernetwork models of binary time series for forecasting price movements on stock markets," in *2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 166–173.
[65] H. Gui, J. Liu, F. Tao, M. Jiang, B. Norick, and J. Han, "Large-scale embedding learning in heterogeneous event data," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, Dec. 2016. [Online]. Available: https://doi.org/10.1109/icdm.2016.0111

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR. Downloaded on November 27,2022 at 19:38:35 UTC from IEEE Xplore. Restrictions apply.