

Stock Selection via Spatiotemporal Hypergraph Attention Network: A Learning to Rank Approach

Ramit Sawhney^{1*}, Shivam Agarwal^{2*}, Arnav Wadhwa³, Tyler Derr⁴, Rajiv Ratn Shah¹

¹IIIT Delhi

²Manipal Institute of Technology

³MIDAS, IIIT-D, India

⁴Vanderbilt University, USA

{ramits, rajivrtn}@iiitd.ac.in, tyler.derr@vanderbilt.edu

Abstract

Quantitative trading and investment decision making are intricate financial tasks that rely on accurate stock selection. Despite advances in deep learning that have made significant progress in the complex and highly stochastic stock prediction problem, modern solutions face two significant limitations. They do not directly optimize the target of investment in terms of profit, and treat each stock as independent from the others, ignoring the rich signals between related stocks' temporal price movements. Building on these limitations, we reformulate stock prediction as a learning to rank problem and propose STHAN-SR, a neural hypergraph architecture for stock selection. The key novelty of our work is the proposal of modeling the complex relations between stocks through a hypergraph and a temporal Hawkes attention mechanism to tailor a new spatiotemporal attention hypergraph network architecture to rank stocks based on profit by jointly modeling stock interdependence and the temporal evolution of their prices. Through experiments on three markets spanning over six years of data, we show that STHAN-SR significantly outperforms state-of-the-art neural stock forecasting methods. We validate our design choices through ablative and exploratory analyses over STHAN-SR's spatial and temporal components and demonstrate its practical applicability.

1 Introduction

The stock market, a financial ecosystem involving transactions between businesses and investors, observed a market capitalization of more than \$68 trillion globally as of the year 2019.¹ Stock trading presents opportunities that increasingly attract traders and investors to utilize the market as a platform for investing and forecasting risk to maximize profit. However, making the right investment decisions and designing trading strategies has many challenges due to the market's highly volatile and non-stationary nature (Adam, Marcket, and Nicoli 2016). Recent advances in deep learning present a promising prospect in quantitative trading and stock prediction (Cavalcante et al. 2016).

However, the vast majority of modern neural stock prediction solutions have two significant drawbacks. First, they

| Stocks: | S ₁ | S ₂ | S ₃ | S ₄ | Performance | Profit(\$) [↑] |
|---|--------------------------------------|----------------|----------------|----------------|--------------------------|-------------------------|
| Returns Ground Truth | +50 ↑ | +30 ↑ | -10 ↓ | -35 ↓ | | |
| Predicted Return (Regression Methods) | R₁ 60 | 10 | 10 | -45 | 250 ↓MSE | 50 ↑ |
| R₂ 25 | 35 | -10 | -40 | 168 ↓ | 30 | |
| Probability of change in return (Classification Methods) | C₁ ↓ 0.6 | ↑ 0.85 | ↓ 0.55 | ↓ 0.8 | 75% ↑ Accuracy | 30 |
| C₂ ↑ 0.8 | ↓ 0.7 | ↑ 0.65 | ↓ 0.75 | 50% ↓ | | 50 ↑ |

Figure 1: Illustration showing that more accurate stock prediction $R_2(\downarrow\text{MSE})$, $C_1(\uparrow\text{Acc.})$ may not always be more profitable than less accurate methods $R_1(\uparrow\text{MSE}), C_2(\downarrow\text{Acc.})$.

are not directly optimized towards the target of investment as they do not factor in the expected earned profit (Feng et al. 2019b). This gap arises because stock prediction is commonly framed either as a classification task to bucket stock movements (price rise, fall or no significant change) or as a regression task to predict stock prices, rather than selecting the stocks with the maximum expected profit (Wang, Wang, and Li 2020; Li et al. 2020). Consider the toy example shown in Figure 1, where we show that methods with high prediction performance do not always lead to the most profitable selection of stocks. Such classification and regression are optimized towards price movement accuracy or minimizing the error in predicting the stock return, and not necessarily towards profit directly. This gap hints towards the disparity between optimizing predictive performance and optimal stock selection for maximizing profit and leads us to think towards a new direction of stock selection, where both predictive performance and the expected profit are jointly and directly optimized.

The second drawback is that the majority of existing work (Feng et al. 2019b; Li et al. 2020; Poli, Park, and Ilievski 2020) treats stock movements to be independent of each other, or utilizes an oversimplified model of the stock market with a graph consisting of pairwise relations between individual stocks when in reality, this is contrary to true market function (Diebold and Yilmaz 2014). Often, stocks are related to each other, and there exist rich signals in the complex higher-order relationships between stocks (or companies) (Nobi et al. 2014; Sawhney et al. 2020b). Publicly available company information can be used to identify connections between stocks that might influence other stocks'

*Equal contribution.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<https://data.worldbank.org/indicator/CM.MKT.LCAP.CD/>

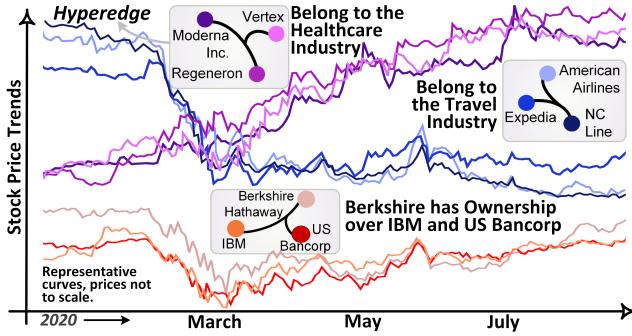


Figure 2: Industry and corporate relations (hyperedges) among stocks can be collectively represented through hypergraphs. Price trends of stocks in a hyperedge are correlated.

prices, such as those stocks having the same CEO or belonging to the same industry. We hypothesize that stocks are often related through higher-order relations as a *collective group*. For instance, Figure 2 shows that related stocks collectively exhibit synchronous price trends. In light of the recent COVID-19 outbreak, we observe that companies belonging to industries like travel and transportation observe a fall in stock prices, as opposed to the observed rise in stock prices of Healthcare related stocks. Hypergraphs being a generalization of graphs, can represent such collective higher-order relations between multiple stocks (companies) simultaneously through hyperedges, as shown in Figure 2.

We formulate stock prediction as a *learning to rank* problem (**Sec. 3.1**), where our model is directly optimized towards *ranking* profitable stocks. We model the higher-order interdependence between stocks as a hypergraph built using *three* different types of stock relations, based on domain knowledge. We then propose STHAN-SR: Spatio Temporal Hypergraph Attention Network for Stock Ranking, a *hypergraph-based* neural architecture for stock prediction. We propose a Hawkes process-based attention mechanism over the temporal trends in stock prices (**Sec. 3.2**). STHAN-SR learns the collective synergy between stocks by combining spatial hypergraph convolutions (**Sec. 3.3**), with a temporal Hawkes attention mechanism through hypergraph attention over stock features to capture the spatial and temporal dependencies in stock movements (**Sec. 3.4**). Through simulations on real-world data of three markets (**Sec. 4.1**), we show that STHAN-SR significantly outperforms state-of-the-art methods in terms of both profit: return, risk-adjusted return: *Sharpe Ratio* and ranking (**Sec. 5**). Lastly, we perform an ablation study (**Sec. 5.2**) and exploratory analysis (**Sec. 5.3, 5.4**) to contextualize each component’s effectiveness and pave the future directions towards more hypergraph problems involving time-evolving features (**Sec. 6**).

The contributions of our work can be summarized as:

- We propose a novel Spatio Temporal Hypergraph Attention Network that models inter stock relations of varying types and degrees as a hypergraph for stock ranking.
- We combine temporal Hawkes attention with spatial hypergraph convolutions through hypergraph attention to

capture correlations in the movements of related stocks and the temporal evolution of their historical features.

- Through experiments on three real-world stock indexes in NYSE, NASDAQ, and TSE markets, over 2,852 stocks spanning over 1,174 trading days, we demonstrate STHAN-SR’s applicability to quantitative stock trading.

2 Related Work

Conventional Methods in Finance Stock movement prediction spans various methods, commonly formulated as regression and classification problems. Financial models conventionally focused on technical analysis and relied only on numerical features (Wang and Leu 1996). Newer models based on Efficient Market Hypothesis are categorized under fundamental analysis, and account for stock affecting factors beyond numerical ones such as financial news, social media, earnings calls, etc. Despite their success, a limitation of these methods is that they assume stock movements to be independent of each other hindering their ability to learn latent patterns for the study of interrelated stocks. A second major limitation is that prior works are not directly optimized for maximizing profit as they do not explicitly select the top stocks with the highest expected revenue.

Contemporary Methods A new line of work revolves around employing graph-based methods to represent pairwise relations between stocks using metadata, such as stock-industry data and links between company CEOs (Sawhney et al. 2020a). For instance, (Kim et al. 2019) propose 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, (Feng et al. 2019b) 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 Learning Hypergraph learning has made progress in problems where relations among data points extend beyond pairwise interactions owing to its ability to extract patterns from higher-order relationships (Feng et al. 2019c). Recent work (Zhang, Zou, and Ma 2019) shows that conventional methods that decompose higher-order relations into a set of pairwise relations do not perform well due to information loss, and that recent methods like Deep Hyper Network Embedding (Tu et al. 2018) are restricted to fixed-length hyperedges leading to poor generalizability.

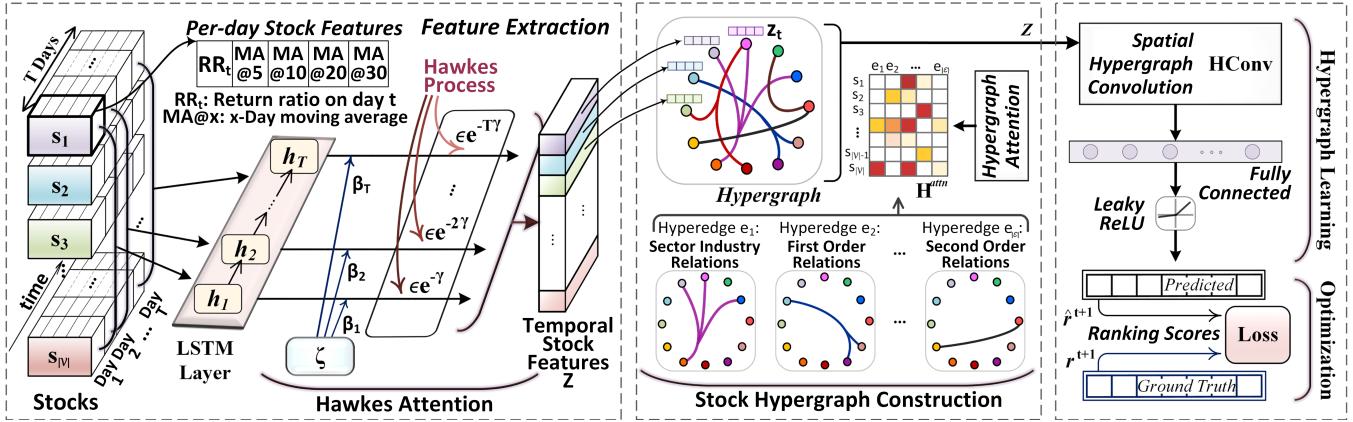


Figure 3: An overall pipeline of the proposed STHAN-SR, stock hypergraph construction, learning, and network optimization.

3 Methodology

3.1 Problem Formulation

We formulate stock prediction as a *learning to rank* problem. Let $S = \{s_1, s_2, \dots, s_N\}$ denote the set of N stocks, where for each stock $s_i \in S$ on trading day t , there is an associated closing price p^t and a 1-day return ratio $r_i^t = \frac{p^t - p^{t-1}}{p^{t-1}}$. On any given trading day t , there exists an optimal ranking $Y^t = \{y_1^t > y_2^t \dots > y_N^t\}$ of the stocks, such that there exists a total order between the ranks $y_i^t > y_j^t$ for any two stocks $s_i, s_j \in S$, if $r_i^t > r_j^t$. Such an ordering of stocks S on a trading day t represents a ranking list, where stocks achieving higher ranking scores Y are expected to achieve a higher investment revenue (profit) on day t . Formally, given stock data for a lookback window of length T (i.e., $[t-T, t-1]$), we aim to learn a ranking function that outputs a score y_i^t to rank each stock s_i on day t in terms of expected profit.

We present an overview of STHAN-SR in Figure 3. In the following subsections, we first show how the price features are extracted, and explain the Hawkes attention mechanism for learning the temporal evolution of stock features (**Sec. 3.2**). We then describe the stock hypergraph construction followed by describing hypergraph convolutions and attention over the stock hypergraph (**Sec. 3.3**). Finally, we combine the temporal and spatial hypergraph components and, optimize the framework to capture temporal and spatial dependencies for end-to-end stock ranking (**Sec. 3.4**).

3.2 Temporal Evolution of Stock Prices

Feature Extraction Historical stock prices have shown to be a strong indicator of future stock trends (Jeanblanc, Yor, and Chesney 2009), and widely used across financial literature (Li et al. 2020; Kim et al. 2019). We use historic price information from previous T trading days as input features to STHAN-SR. We calculate five temporal features for each stock, 1-day return ratio, 5, 10, 20 and 30 day moving averages which represent the daily, weekly and monthly trends. For each stock s , we then concatenate these temporal features to form a stock price feature vector q_τ on day τ . We then use an LSTM to capture the temporal dependencies in

stock features q_τ . We feed the daily price features q_τ of each stock to obtain the hidden states $h_\tau \in \mathbb{R}^d$ for day τ as:

$$h_\tau = \text{LSTM}(q_\tau, h_{\tau-1}), \quad t-T \leq \tau \leq t-1 \quad (1)$$

where, d represents the dimension of the hidden states.

Temporal Attention Studies show that the stock trend of each day has a different impact on future prices. To this end, we employ a temporal attention mechanism $\zeta(\cdot)$ which learns to weigh critical days that impact future prices. This mechanism aggregates temporal hidden states $\bar{h}_t = [h_{t-T}, \dots, h_{t-1}] \in \mathbb{R}^{d \times T}$ from different days into an overall representation using learnt attention weights β_τ for each day τ . We formulate this mechanism as:

$$\zeta(\bar{h}_t) = \sum_{\tau} \lambda_{\tau}, \quad \lambda_{\tau} = \beta_{\tau} h_{\tau}, \quad \beta_{\tau} = \frac{\exp(h_{\tau}^T W \bar{h}_t)}{\sum_{\tau} \exp(h_{\tau}^T W \bar{h}_t)} \quad (2)$$

where, W is a learned linear transform. β_{τ} represents the learnt attention weights used to aggregate all temporal features while assigning higher weights to important features.

Hawkes Attention The Hawkes process is a temporal point counting-process that models a sequence of arrival of events over time. Each event “excites” the process in the sense that the chance of a subsequent arrival is increased for some time. In stock markets, events such as release of earning call statements, crises situations etc. influence the future prices and such influence decays over time. It has been shown in financial literature that Hawkes process can be used to model historic stock prices and predict future trends (Bacry, Mastromatteo, and Muzy 2015). We propose a temporal Hawkes attention mechanism which enhances the temporal attention mechanism $\zeta(\cdot)$ by using a Hawkes process while aggregating day level latent representations λ_{τ} . This attention mechanism learns an excitation parameter ϵ corresponding to day τ and a decay parameter γ to learn the decay rate of this induced excitement. For each stock, we compute a temporal feature z_t as:

$$z_t = \sum_{\tau=0, \Delta t_{\tau} \geq 0} (\lambda_{\tau} + \epsilon \max(\lambda_{\tau}, 0) e^{-\gamma \Delta t_{\tau}}) \quad (3)$$

Δt_{τ} is the time difference between current and past day τ . We concatenate features z_t of all stocks to form $\mathbf{Z} \in \mathbb{R}^{N \times d}$.

3.3 Spatial Stock Hypergraph Feature Extraction

Stock Hypergraph Construction We model stock interdependence via hypergraphs, where hyperedges represent higher-order relations between stocks. We construct 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 a subset of related stocks $\{s_1, s_2, \dots, s_n\} \in S$. Each hyperedge e is assigned a positive weight $w(e)$ with all weights stored in a diagonal matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$. We let $\mathbf{W} = \mathbf{I}$ indicating equal weights for all hyperedges. We inject domain knowledge by constructing hyperedges between stocks based on two types of relations: industry and Wiki corporate relations.

Industry Hyperedges: Stocks belonging to same industries, collectively experience similar price trends based on the industry's performance (Livingston 1977). To leverage this signal, we define relations between stocks as per the GICS standard.² Formally, we construct a hyperedge $e \in \mathcal{E}_{ind}$ that connects stocks that belong to the same industry.

Wiki Corporate Hyperedges: We consider two types of corporate relationships between stocks (companies) based on Wikidata (Vrandečić and Krötzsch 2014). The first type of corporate relation is a first-order relation, which is defined as $X \xrightarrow{R1} Y$ where R1 represents the entity-relation between stocks X and Y defined in Wikidata. As shown in Figure 3, we use these relationships to construct a hyperedge $e \in \mathcal{E}_1$ which consists a source stock and a set of target stocks related to the source stock through the same Wikidata relation. For instance, BlackRock and companies it owns (Netflix, eBay, Phillips 66, etc.) are represented by a hyperedge.

The second type of corporate relation is a second order relationship, which is pairwise in nature. This relation is defined as $X \xrightarrow{R2} Z \xleftarrow{R3} Y$ where Z denotes an entity connecting the two stocks X and Y via entity-relations R2, and R3. We construct a hyperedge $e \in \mathcal{E}_2$ between two stocks in a second order relationship. For instance, Microsoft(X) and Berkshire Hathaway(Y) are related through Bill Gates(Z) since he owns Microsoft(R2: “owned by”), and is a board member of Berkshire Hathaway (R3: “is a board member of”). We combine these relations as $\mathcal{E} = \mathcal{E}_{ind} \cup \mathcal{E}_1 \cup \mathcal{E}_2$ to construct the hypergraph \mathcal{G} , equivalently denoted by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, with entries $h(v, e)$ defined as:

$$h(v, e) = \begin{cases} 1, & v \in e \\ 0, & v \notin e \end{cases} \quad (4)$$

The degree of each vertex v is obtained using a function $d(v)$, and stored in a diagonal matrix $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ as:

$$d(v) = \sum_{e \in \mathcal{E}} w(e) h(v, e) = \sum_{e \in \mathcal{E}} h(v, e) \quad (5)$$

The degree of each hyperedge e is obtained using $\delta(e) = \sum_{v \in \mathcal{V}} h(v, e)$ stored in a diagonal matrix $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$.

Hypergraph Convolution To learn the interdependence between the price movements of stocks, we use a hypergraph convolution (Figure 4) on the hypergraph \mathcal{G} (Gilmer

²wikipedia.org/wiki/Global_Industry_Classification_Standard

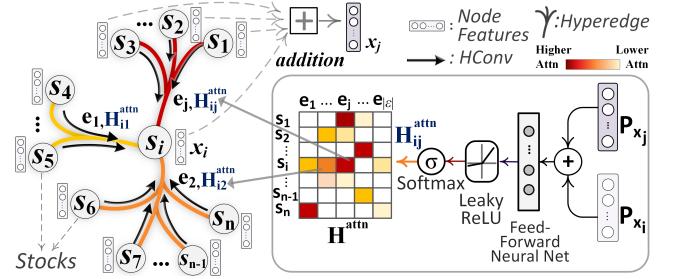


Figure 4: Hypergraph convolutions among stocks in a hyperedge (left) and attention among nodes & hyperedges (right).

et al. 2017). We first define a single hypergraph convolution $HConv(\cdot)$, where the input to the l^{th} hypergraph convolution layer is a matrix of temporal stock features $\mathbf{X}^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times F^{(l)}}$ where $F^{(l)}$ is the dimension of the temporal features. The hypergraph convolution updates the temporal features $\mathbf{X}^{(l)}$ to new features $\mathbf{X}^{(l+1)} \in \mathbb{R}^{|\mathcal{V}| \times F^{(l+1)}}$ using the neighboring stock features with their structural relationships represented in the hypergraph \mathcal{G} where, $F^{(l+1)}$ is the dimension of transformed node features. Following (Feng et al. 2019c), we define the hypergraph Laplacian as $\Delta = \mathbf{I} - \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-1/2}$ resulting in the hypergraph convolution update rule as:

$$\begin{aligned} \mathbf{X}^{(l+1)} &= HConv(\mathbf{X}^{(l)}, \mathbf{H}, \mathbf{P}) \\ &= \text{ELU}(\mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{X}^{(l)} \mathbf{P}) \end{aligned} \quad (6)$$

$\mathbf{P} \in \mathbb{R}^{F^{(l)} \times F^{(l+1)}}$ is a learnable matrix, ELU is exponential linear unit activation. Recall that we set $\mathbf{W} = \mathbf{I}$.

Hypergraph Attention To capture the varying degree of influence each stock relation has on the temporal price evolution of each stock, we employ a hypergraph attention mechanism on the incidence matrix \mathbf{H} (Bai, Zhang, and Torr 2019). This mechanism learns to adaptively weight each hyperedge associated with a stock based on its temporal features, adding a learning mechanism over the stock relations, thereby bridging the temporal Hawkes attention and spatial hypergraph convolutions. Formally, for each node $v_i \in \mathcal{V}$ and its associated hyperedge $e_j \in \mathcal{E}$, we compute an attention coefficient H_{ij}^{attn} using the stock's temporal feature x_i and the aggregated hyperedge features x_j , quantifying how important the corresponding relation e_j is to the stock v_i . Formally, we define the attention coefficient as the softmax of a single-layer feed forward network \vec{a} :

$$H_{ij}^{\text{attn}} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [\mathbf{P}x_i \oplus \mathbf{P}x_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\vec{a}^T [\mathbf{P}x_i \oplus \mathbf{P}x_k]))} \quad (7)$$

where, \oplus is concatenation and \mathbf{P} represents a learnt linear transform. \mathcal{N}_i is the neighborhood set of x_i which can be accessed using the constructed hypergraph \mathcal{G} . The attention-based learnt hypergraph incidence matrix \mathbf{H}^{attn} shown in Figure 4, is then used in the above hypergraph convolution

| | NASDAQ | NYSE | TSE |
|------------------|---------------|-------------|-------------|
| Train(Tr) Period | 01/13-12/15 | 01/13-12/15 | 11/15-08/18 |
| Val(V) Period | 01/16-12/16 | 01/16-12/16 | 08/18-07/19 |
| Test(Te) Period | 01/17-12/17 | 01/17-12/17 | 07/19-08/20 |
| #Days(Tr:V:Te) | 756:252:237 | 756:252:237 | 693:231:235 |
| #Stocks(Nodes) | 1,026 | 1,737 | 95 |
| #Hyperedges | 862 | 1,595 | 84 |

Table 1: Dataset statistics detailing chronological date splits of the three markets and their corresponding hypergraphs.

shown in Equation 6 by replacing \mathbf{H} with \mathbf{H}^{at} to learn intermediate representations of the stocks (nodes) layer-by-layer.

We use multi-headed architecture to stabilise training (Vaswani et al. 2017). Formally, K independent executors apply the hypergraph convolution using the enriched incidence matrix \mathbf{H}^{at} whose outputs are concatenated to yield:

$$\mathbf{X}^{(l+1)} = \bigoplus_{k=1}^K \text{ELU}\left(\mathbf{D}_v^{-\frac{1}{2}} \mathbf{H}_k^{\text{at}} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}_k^{\text{at}^T} \mathbf{D}_v^{-\frac{1}{2}} \mathbf{X}^{(l)} \mathbf{P}_k\right) \quad (8)$$

where, \mathbf{H}_k^{at} and \mathbf{P}_k are the enriched incidence matrix and the weight matrix of the k^{th} executor, respectively.

3.4 Learning to Rank and Network Optimization

We employ two hypergraph convolutions ($\text{HConv}(\cdot)$) with an ELU activation between the first and second layer. We feed the temporal stock features \mathbf{Z} to STHAN-SR’s first layer $\mathbf{X}^{(0)} \in \mathbb{R}^{|\mathcal{V}| \times d}$. STHAN-SR’s final layer $\mathbf{X}^{(2)} \in \mathbb{R}^{|\mathcal{V}| \times 1}$ outputs the predicted stock ranking $\hat{\mathbf{r}}^{t+1}$.

$$\hat{\mathbf{r}}^{t+1} = \text{HConv}\left(\bigoplus_{k=1}^K \text{HConv}(\mathbf{Z}, \mathbf{H}^{\text{at}}, \mathbf{P}_1), \mathbf{H}^{\text{at}}, \mathbf{P}_2\right) \quad (9)$$

\mathbf{P}_1 and \mathbf{P}_2 are parameter matrices of first and second layers.

We optimize STHAN-SR using a combination of a pointwise regression and pairwise ranking-aware loss to minimize the difference between the predicted and actual return ratios while maintaining the relative order of top ranked stocks with higher expected return for investment as:

$$L = \|\hat{\mathbf{r}}^{t+1} - \mathbf{r}^{t+1}\|^2 + \phi \sum_{i=0}^{|\mathcal{V}|} \sum_{j=0}^{|\mathcal{V}|} \max\left(0, -\left(\hat{r}_i^{t+1} - \hat{r}_j^{t+1}\right)\left(r_i^{t+1} - r_j^{t+1}\right)\right) \quad (10)$$

where, \hat{r}^{t+1} and r^{t+1} are the predicted and actual ranking scores, respectively, and ϕ is a weighting parameter.

4 Experimental Setup

4.1 Datasets

For a comprehensive evaluation of STHAN-SR, we evaluate it on *three* real-world datasets from *US* and *Japanese* stock markets spanning over *six* years. We summarize statistics about the datasets in Table 1, and elaborate on them next:

NASDAQ (Feng et al. 2019b) is a fairly volatile US exchange. We evaluate STHAN-SR on 1,026 equity stocks from the NASDAQ Global and Capital markets that span the S&P 500 and NASDAQ Composite indexes.

NYSE (Feng et al. 2019b) is the world’s largest stock exchange by market capitalization and is stable as compared to NASDAQ.

Tokyo Stock Exchange (TSE) (Li et al. 2020) is a smaller market contrasting with US markets. We evaluate STHAN-SR on the 95 largest stocks by market capitalization in Japan, in the TOPIX 100 index.

We collect prices using Google Finance³ and construct hypergraphs by mining relations from Wikidata.⁴

4.2 Training Setup

We perform all experiments on a Tesla P100 GPU. We use grid search to find optimal hyperparameters, lookback length $T \in \text{range}[8, 24]$, weighting factor $\phi \in \text{range}[1, 10]$ and learning rate $\in (1e-4, 5e-3)$ for all models based on validation Normalized Discounted Cumulative Gain (NDCG@5). We use Xavier initialization for all weights and set LSTM output space to 32. We set number of attention heads $K = 4$ and output space of $\text{HConv}(\cdot)$ to 32. We use Adam optimizer and train STHAN-SR for 500 epochs.

4.3 Evaluation Metrics

Returns We compare the Sharpe ratio and cumulative investment return ratio (IRR) to assess profit generation ability of all methods. Following (Feng et al. 2019b), we adopt a daily buy-hold-sell trading strategy that is, when the market closes on trading day t the trader uses the method to get a ranked list of the predicted return ratio for each stock. The trader then buys the top k stocks and then sells the bought stocks on the market close of trading day $t + 1$. The IRR is thus, the cumulative return on an investment over time, independent of the length of the duration. The IRR on day t is defined as, $\text{IRR}^t = \sum_{i \in \mathcal{S}^{t-1}} \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$ where, \mathcal{S}^{t-1} denotes the set of stocks in the portfolio on day $t - 1$ and p_i^t , p_i^{t-1} is the closing price of the stock i on day t and $t - 1$ respectively. We also calculate the **Sharpe ratio (SR)**, which is a measure of the return of a portfolio compared to its risk. We calculate the Sharpe ratio by computing the earned return R_a in excess of a risk-free return⁵ R_f as:

$$\text{Sharpe Ratio}_a = \frac{E[R_a - R_f]}{\text{std}[R_a - R_f]} \quad (11)$$

Ranking We evaluate STHAN-SR’s ranking ability using NDCG@ k . NDCG@ k sums the true scores ranked in the order induced by the predicted scores, after applying logarithmic discount. For both returns and NDCG, we report results for top $k = 5$ stocks.

5 Results and Analysis

5.1 Profitability Comparison with Baselines

We compare STHAN-SR with several baselines in terms of profitability, as shown in Table 2. We observe that STHAN-SR consistently generates significantly ($p < 0.01$) higher

³Google Finance: <https://www.google.com/finance>

⁴Wikidata: <https://www.wikidata.org/>

⁵T-Bill rates: <https://home.treasury.gov/>

| Model | Description | NASDAQ | | NYSE | | TSE | |
|---------------------------------|---|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | | SR↑ | IRR↑ | SR↑ | IRR↑ | SR↑ | IRR↑ |
| ARIMA (Wang and Leu 1996) | Non-stationary time series modeling of prices | 0.55 | 0.10 | 0.33 | 0.10 | 0.47 | 0.13 |
| A-LSTM (Feng et al. 2019a) | Adversarial training to enhance stock prediction | 0.97 | 0.23 | 0.81 | 0.14 | 1.10 | 0.43 |
| CLF HGCluster (Luo et al. 2014) | Graph coarsening on price correlation hypergraph | 0.06 | 0.10 | 0.10 | 0.11 | 0.20 | 0.10 |
| GCN (Li et al. 2020) | LSTM+GCN for modelling stock relation graph | 0.75 | 0.13 | 0.70 | 0.10 | 0.90 | 0.28 |
| HATS (Kim et al. 2019) | Hierarchical graph attention on stock multigraphs | 0.80 | 0.15 | 0.73 | 0.12 | 0.96 | 0.31 |
| REG SFM (Zhang et al. 2017) | Frequency-aware LSTM on price data | 0.16 | 0.09 | 0.19 | 0.11 | 0.08 | 0.07 |
| LSTM (Bao et al. 2017) | Vanilla LSTM on temporal price data | 0.48 | 0.13 | 0.13 | 0.09 | 0.63 | 0.20 |
| RL DQN (Carta et al. 2020) | Ensemble of deep Q-learning agents on price data | 0.93 | 0.20 | 0.72 | 0.12 | 1.08 | 0.31 |
| iRDPG (Liu et al. 2020) | Imitative RDPG algorithm on temporal price data | 1.32 | 0.28 | 0.85 | 0.18 | 1.10 [†] | 0.55 [†] |
| LSTM (Bao et al. 2017) | Vanilla LSTM on temporal price data for ranking | 0.95 | 0.22 | 0.79 | 0.12 | 0.73 | 0.21 |
| GCN (Kipf et al. 2017) | LSTM+GCN for modelling stock relation graph | 0.46 | 0.13 | 0.72 | 0.16 | 0.81 | 0.27 |
| RAN RSR-E (Feng et al. 2019b) | Temporal GCN using similarity as relation weight | 1.12 | 0.26 | 0.88 | 0.20 | 1.07 | 0.50 |
| RSR-I (Feng et al. 2019b) | Temporal GCN with neural net for relation weight | 1.34* | 0.39* | 0.95* | 0.21* | 1.08 | 0.53 |
| STHAN-SR (Ours) | Hawkes Attention, HG Attention, HG Convolution | 1.42 ^{*†} | 0.44 ^{*†} | 1.12 ^{*†} | 0.33 ^{*†} | 1.19 ^{*†} | 0.62 ^{*†} |

Table 2: Profitability comparison with classification (CLF), regression (REG), reinforcement learning (RL) and ranking (RAN) methods (mean of 5 individual runs). Bold & italics show best & second best (SOTA) results, respectively. * & † imply the improvement over iRDPG & RSR-I, respectively, is statistically significant ($p < 0.01$), under Wilcoxon’s signed rank test.

| Ablation Study | NASDAQ | | | NYSE | | | TSE | | |
|-------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | SR↑ | IRR↑ | NDCG↑ | SR↑ | IRR↑ | NDCG↑ | SR↑ | IRR↑ | NDCG↑ |
| Model Component | | | | | | | | | |
| LSTM | 0.95 | 0.22 | 0.68 | 0.79 | 0.12 | 0.65 | 0.73 | 0.21 | 0.65 |
| Temporal Attention + LSTM | 0.96* | 0.29* | 0.67 | 0.80* | 0.13* | 0.67* | 0.80* | 0.23* | 0.70* |
| Hawkes Attention + LSTM | 1.06* | 0.31* | 0.69* | 0.96* | 0.17* | 0.68* | 0.86* | 0.27* | 0.72* |
| Hypergraph Conv + LSTM | 0.93 | 0.27* | 0.65 | 0.78 | 0.15* | 0.67* | 0.82* | 0.28* | 0.71* |
| Hypergraph Conv + Hawkes Attn | 1.00* | 0.35* | 0.71* | 0.81* | 0.20* | 0.70* [†] | 0.97* | 0.34* | 0.73* |
| Hypergraph Attn + LSTM | 1.37* [†] | 0.42* [†] | 0.74* [†] | 0.98* [†] | 0.24* [†] | 0.72* [†] | 1.11* [†] | 0.56* [†] | 0.76* [†] |
| STHAN-SR | 1.42* [†] | 0.44* [†] | 0.80* [†] | 1.12* [†] | 0.33* [†] | 0.88* [†] | 1.19* [†] | 0.62* [†] | 0.84* [†] |

Table 3: Ablation study over STHAN-SR’s components (mean of 5 independent runs). * and † indicate improvement over the LSTM and state-of-the-art RSR-I, respectively, is statistically significant ($p < 0.01$), under Wilcoxon’s signed rank test.

risk-adjusted returns than all baselines across all datasets. Generally, ranking and RL methods that are inherently optimized for higher returns are more profitable than classification and regression methods, which do not necessarily select the most profitable stocks to trade, validating our premise of formulating stock prediction as a learning to rank problem.

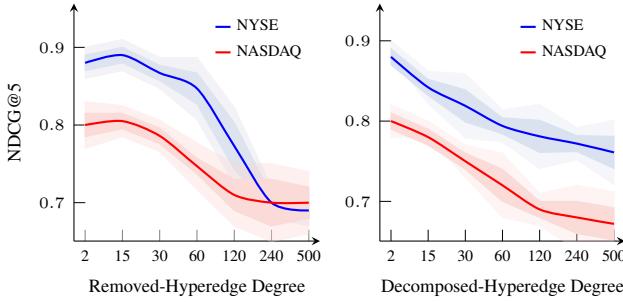
Next, we observe that amongst the best performing ranking and RL models, those that model stock interdependence (RSR-I, STHAN-SR) outperform price-only methods (LSTM, DQN, iRDPG), as they capture the spatial correlations amongst movements of related stocks. We also observe that our proposed STHAN-SR significantly ($p < 0.01$) outperforms state-of-the-art graph-based methods (RSR-I, RSR-E). We postulate this improvement to STHAN-SR’s design that captures higher-order stock relations as a hypergraph instead of constraining them as pairwise edges in ordinary graphs (GCN, RSR-I, and RSR-E). These observations collectively demonstrate STHAN-SR’s utility as a spatiotemporal attention-based hypergraph learning to rank stock selection model. We now further probe into each of STHAN-SR’s components through an ablation study to analyze the sources of these improvements over the baselines.

5.2 Model Component Ablation Study

In Table 3, we observe that Hawkes attention significantly ($p < 0.01$) improves temporal attention, validating our design choice of using Hawkes process to model stock prices as temporal point processes. On the spatial front, we note that hypergraph convolutions over inter stock relations do not lead to significant improvements, likely because from the vast number of diverse relations between stocks, only a few are meaningful enough to significantly influence the prices of related stocks. Intuitively, complementing hypergraph convolutions with attention leads to large improvements, as the spatial hypergraph attention mechanism learns to weigh more important relations selectively, such as one where a set of stocks have the same parent company, as opposed to stocks related by their country of origin. Finally, we observe that both spatial hypergraph and temporal Hawkes components complement each other by capturing spatiotemporal correlations in stock markets. Next, we investigate both the spatial and temporal components to further contextualize performance improvements due to each component.

5.3 On the Effectiveness of Hypergraphs

Effect of injecting domain knowledge via stock relations
We first probe the effectiveness of domain knowledge-based



(a) Successive removal of hyperedges from the hypergraph (b) Decomposition of hyperedges into pair-wise relations

Figure 5: Influence of domain knowledge via hypergraphs.

hyperedges, by successively removing them in an increasing order of their degree and analyzing STHAN-SR’s ranking performance in terms of NDCG@5 in Figure 5a. We observe that NDCG@5 decreases as we remove hyperedges, and performs the worst after all hyperedges are removed, essentially degenerating STHAN-SR to a Hawkes Attention + LSTM model that does not account for inter-stock relations.

Hypergraph v.s. Graph for representing stock relations
We now contrast the performance between representing inter stock relations as hyperedges as opposed to ordinary pairwise edges. We decompose each hyperedge of degree n into $\binom{n}{2}$ pairwise edges, in an increasing order of hyperedge degree and analyze the NDCG@5 variation as we decompose hyperedges in Figure 5b. We observe a degradation in ranking ability as we decompose hyperedges into pairwise edges, with the minimum NDCG@5 being attained when all hyperedges are decomposed, essentially when STHAN-SR degrades into a Hawkes Attention + Graph Attention Network.

Through these experiments, we note that modeling inter stock dependence through domain knowledge as (hyper)edges drastically improves stock selection, and more importantly, that hypergraphs effectively capture higher order relations between stocks, as opposed to simple graphs.

5.4 Visualizing Hawkes Attention

Next, we qualitatively analyze STHAN-SR’s temporal component by comparing Hawkes (HA) and temporal attention (TA) mechanisms over a 16-day lookback for the stock USAP from NASDAQ’s test set in March 2017. We visualize day-level attention throughout the lookback window, and analyze the corresponding predicted Return Ratios (RR) for the 17th day in Figure 6. For comparison, we also show actual and previously predicted RRs across all days. STHAN-SR using TA predicts the 17th day RR with a relative error of 5.57% from the actual value, whereas using HA, predicts a return closer to the actual value (0.69%). Despite the varying trend throughout the lookback window, HA accurately captures the uptrend towards the end of the window, whereas TA learns distributed scores, capturing an overall downtrend.

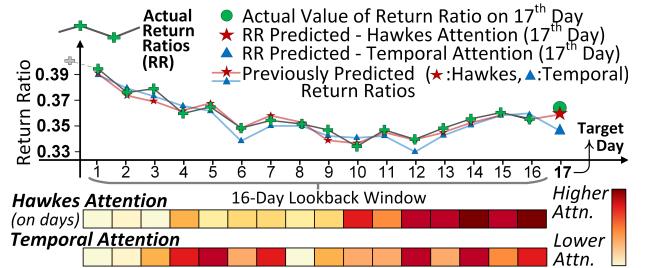


Figure 6: Day-level attention visualization and returns using STHAN-SR with Hawkes and temporal attention methods.

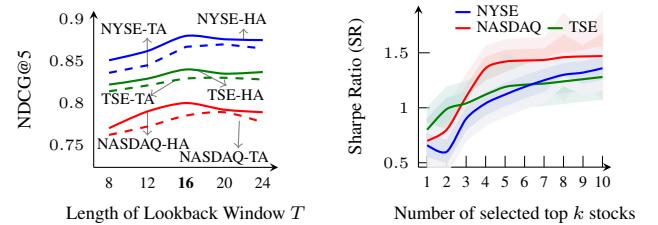


Figure 7: Sensitivity to parameters T and k

5.5 Parameter Analysis: Probing Sensitivity

Lookback window length T We analyze STHAN-SR’s ranking performance with varying historical lookback lengths T in Figure 7 and observe that STHAN-SR using Hawkes attention outperforms temporal attention, and performs better over longer lookbacks.

Number of selected top stocks k We analyze STHAN-SR’s profitability (SR) variation with the number of selected top stocks k from the ranked stocks in Figure 7. We find that STHAN-SR performs well and is consistent on varying k .

6 Conclusion and Future Work

We reformulate stock prediction as a *learning to rank* problem and model stocks via *hypergraphs* based on domain knowledge. We present STHAN-SR, a neural hypergraph model for stock prediction. We propose temporal Hawkes attention and complement it with spatial hypergraph convolutions and attention to capture the spatiotemporal dependencies in stock markets. STHAN-SR significantly outperforms state-of-the-art methods in terms of profit in three global markets over six years. Through ablative and qualitative experiments, we probe STHAN-SR’s effectiveness and set forth its practical applicability for algorithmic trading. Our proposed model can be generalized for spatiotemporal learning over hypergraphs across problems in varying domains, such as traffic prediction and session-based recommender systems. In future, we aim to explore time-evolving hypergraphs to capture dynamic market correlations and incorporate additional data sources such as online news and social media.

References

- Adam, K.; Marcket, A.; and Nicoli, J. P. 2016. Stock Market Volatility and Learning. *The Journal of Finance* 71(1): 33–82.
- Bacry, E.; Mastromatteo, I.; and Muzy, J.-F. 2015. Hawkes Processes in Finance. *Market Microstructure and Liquidity* 01(01): 1550005. ISSN 2424-8037.
- Bai, S.; Zhang, F.; and Torr, P. H. S. 2019. Hypergraph Convolution and Hypergraph Attention.
- 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*.
- Carta, S.; Ferreira, A.; Podda, A. S.; Recupero, D. R.; and Sanna, A. 2020. Multi-DQN: an Ensemble of Deep Q-Learning Agents for Stock Market Forecasting. *Expert Systems with Applications* 113820.
- Cavalcante, R. C.; Brasileiro, R. C.; Souza, V. L.; Nobrega, J. P.; and Oliveira, A. L. 2016. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications* 55: 194–211.
- Diebold, F.; and Yilmaz, K. 2014. On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics*.
- Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; and Chua, T.-S. 2019a. Enhancing stock movement prediction with adversarial training. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*.
- 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, F.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019c. Hypergraph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33: 3558–3565.
- Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry.
- Jeanblanc, M.; Yor, M.; and Chesney, M. 2009. *Mathematical methods for financial markets*. Springer Science & Business Media.
- 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:1908.07999*.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Li, W.; Bao, R.; Harimoto, K.; Chen, D.; Xu, J.; and Su, Q. 2020. Modeling the Stock Relation with Graph Network for Overnight Stock Movement Prediction. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 4541–4547.
- Liu, Y.; Liu, Q.; Zhao, H.; Pan, Z.; and Liu, C. 2020. Adaptive Quantitative Trading: An Imitative Deep Reinforcement Learning Approach. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(02): 2128–2135.
- Livingston, M. 1977. Industry movements of common stocks. *The Journal of Finance* 32(3): 861–874.
- Luo, Y.; Hu, J.; Wei, X.; Fang, D.; and Shao, H. 2014. Stock trends prediction based on hypergraph modeling clustering algorithm. In *2014 IEEE International Conference on Progress in Informatics and Computing*.
- Nobi, A.; Maeng, S. E.; Ha, G. G.; and Lee, J. W. 2014. Effects of global financial crisis on network structure in a local stock market. *Physica A: Statistical Mechanics and its Applications* ISSN 0378-4371.
- Poli, M.; Park, J.; and Ilievski, I. 2020. WATTNet: Learning to Trade FX via Hierarchical Spatio-Temporal Representation of Highly Multivariate Time Series. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 4569–4575.
- Sawhney, R.; Agarwal, S.; Wadhwa, A.; and Shah, R. R. 2020a. Deep Attentive Learning for Stock Movement Prediction From Social Media Text and Company Correlations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 8415–8426. Online: Association for Computational Linguistics.
- Sawhney, R.; Agarwal, S.; Wadhwa, A.; and Shah, R. R. 2020b. Spatiotemporal Hypergraph Convolution Network for Stock Movement Forecasting. In *2020 IEEE International Conference on Data Mining (ICDM)*, 482–491. IEEE.
- Tu, K.; Cui, P.; Wang, X.; Wang, F.; and Zhu, W. 2018. Structural deep embedding for hyper-networks. In *AAAI*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 78–85.
- Wang, H.; Wang, T.; and Li, Y. 2020. Incorporating Expert-Based Investment Opinion Signals in Stock Prediction: A Deep Learning Framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 971–978.
- Wang, J.-H.; and Leu, J.-Y. 1996. Stock market trend prediction using ARIMA-based neural networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 4, 2160–2165. IEEE.
- Zhang, L.; Aggarwal, C.; and Qi, G.-J. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2141–2149.
- Zhang, R.; Zou, Y.; and Ma, J. 2019. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*.