Task 2

Chaithra

January 2025

1 Task 2 Details

The Task 2 overview is shown in Figure 1. Task 2 requires the preparation of weekly data from daily data and then feeding the weekly data to attentive GRU, which generates the weekly embedding, which is then fed to the GAT model, which generates the node embedding based on neighbor node representation. The past t week weekly embedding from GRU and the graph node embedding from GAT model is fed to Attention Learning for long term sequential learning.

Reference Paper: FinGAT: Financial Graph Attention Networks for Recommending Top-K Profitable Stocks

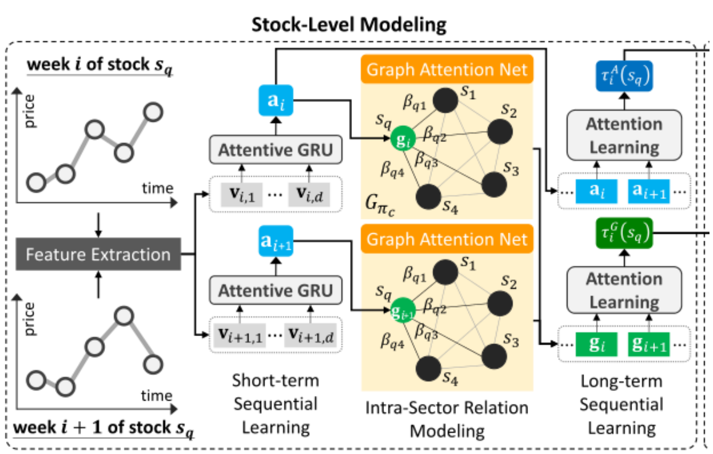

Figure 1: Task2

For task 2, ensure that the companies considered under the Nifty 500 have the same years of data. If any company has fewer years of data, that company should not be considered. ONGC has 1236 rows of data.

Total number of companies with less data: 93

Company	**Company**	**Company**	**Company**
AADHARHFC: 167 rows	ADANIENSOL: 344 rows	AWL: 724 rows	ABSLAMC: 806 rows
AEGISLOG: 154 rows	AKUMS: 108 rows	ARE&M: 301 rows	ANANDRATHI: 763 rows
ANGELONE: 1059 rows	APTUS: 839 rows	ACI: 531 rows	BHARTIHEXA: 188 rows
BIKAJI: 534 rows	MAPMYINDIA: 758 rows	CAMPUS: 665 rows	CELLO: 293 rows
CHEMPLASTS: 839 rows	CLEAN: 863 rows	CAMS: 1059 rows	CONCORDBIO: 346 rows
CRAFTSMAN: 940 rows	DOMS: 263 rows	DATAPATTNS: 755 rows	DELHIVERY: 654 rows
DEVYANI: 844 rows	DUMMYITC: 0 rows	EASEMYTRIP: 944 rows	EMCURE: 126 rows
EQUITASBNK: 1039 rows	NYKAA: 786 rows	FIVESTAR: 531 rows	GRINFRA: 863 rows
GLAND: 1025 rows	MEDANTA: 532 rows	GODIGIT: 162 rows	HBLENGINE: 23 rows
HAPPSTMNDS: 1070 rows	POWERINDIA: 1187 rows	HOMEFIRST: 975 rows	HONASA: 292 rows
INOXINDIA: 262 rows	INDGN: 169 rows	IRFC: 978 rows	IREDA: 278 rows
JSWINFRA: 312 rows	JIOFIN: 341 rows	JUBLINGREA: 944 rows	JYOTICNC: 245 rows
KPIL: 385 rows	KALYANKJIL: 939 rows	KAYNES: 530 rows	KFINTECH: 503 rows
KIMS: 878 rows	LATENTVIEW: 778 rows	LICI: 659 rows	LLOYDSME: 369 rows
LODHA: 926 rows	MANKIND: 417 rows	MAXHEALTH: 1089 rows	MAZDOCK: 1054 rows
METROBRAND: 757 rows	MSUMI: 692 rows	NSLNISP: 467 rows	NETWEB: 361 rows
NUVAMA: 316 rows	NUVOCO: 840 rows	PAYTM: 780 rows	POLICYBZR: 783 rows
PTCIL: 394 rows	PVRINOX: 519 rows	PPLPHARMA: 552 rows	RRKABEL: 321 rows
RAILTEL: 958 rows	RAINBOW: 664 rows	ROUTE: 1068 rows	SBFC: 344 rows
SBICARD: 1197 rows	SAPPHIRE: 780 rows	SHYAMMETL: 880 rows	SIGNATURE: 319 rows
SONACOMS: 880 rows	STARHEALTH: 765 rows	SUMICHEM: 1230 rows	SUVENPHAR: 1201 rows
SYRMA: 588 rows	TBOTEK: 167 rows	TVSSCS: 339 rows	TATATECH: 273 rows
UTIAMC: 1053 rows	UNITDSPR: 154 rows	MANYAVAR: 718 rows	VIJAYA: 825 rows
ZOMATO: 860 rows			

1.1 Train, Validation, Testing Days

The 5-year data has to be split as follows: A typical year has 252 trading days,this is excluding the holidays Training Data : 3 Years (756 days) Validation Data :1 Year (252 days) Testing Data :1 Year (252 days)

1.2 Parameter Settings

Note: The number of weeks to be considered for the short-term sequential learning and long-term learning should vary as $w \in \{1, 2, 3, 4\}$. The dimensions for hidden layers of GRU and GAT must be varied for $\{8, 16, 32, 64\}$

2 Implementation

The task2 requires the implementation of the following:

2.1 Data Preparation

Data has to prepared for training ,validation and test data separately.

The daily data features are:

1. Open

2. Close
3. High
4. Low
5. return ratio
6. percentage change in open
7. percentage change in high
8. percentage change in low
9. Moving average (5 days)
10. Moving average(10 days)
11. Moving average(15 days)
12. Moving average(20 days)
13. Moving average(25 days)
14. Moving average(30 days)
15. Sector Information

The daily data has to be transformed into weekly data using sliding window approach where each week consists of 5 trading days (as there are 5 trading days in a week). The sliding window technique is a method for analyzing time series data by dividing it into overlapping windows and processing each window individually. The number of weeks can vary (1, 2, 3, or 4 weeks), which corresponds to 5, 10, 15, or 20 days, respectively.

For each week of data, the labels are calculated as:

Return ratio: The return for the 6th day after the window of the weekly data. For example, if we are using a 5-day window (1 week), the return ratio for the 6th day is the the label. Stock movement label: This represents the movement of the stock on the 6th day after the window. For example, if the stock price goes up on the 6th day, the stock movement label could be classified as "up" or "1," and if the stock price goes down, it could be classified as "down" or "0."

```
|  Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | --> Label (Day 6)
      Slide -->
|  Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | --> Label (Day 7)
```

Figure 2: Sliding Window

2.2 Short-Term Sequential Learning

Input : The Week w=1,2,3,4 data ie 5,10,15,20 day data.
Output: Weekly Embeddings through GRU ie 8,16,32,64

Description: The short-term sequential learning module leverages an Attentive GRU to process daily time-series data for each stock. For a given stock S_q, the features corresponding to week i are represented as $V_{i,j}^{sq}$, where j denotes the j^{th} day of the week.

These daily feature vectors capture essential information about the stock's behavior and are sequentially passed into the GRU, as shown in Figure 3.

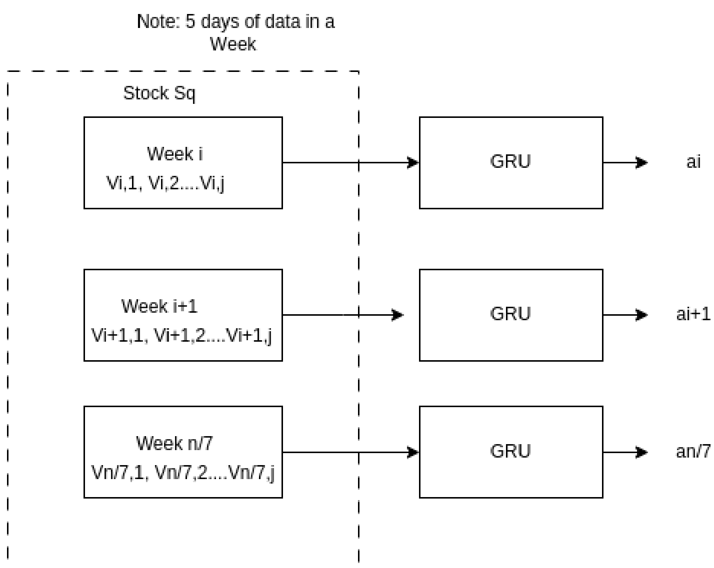

Figure 3: Short-Term Sequential Learning using Attentive GRU

The feature vector $V_{i,j}^{sq}$ is first mapped to a 64-dimensional embedding space by passing it through a GRU layer. This GRU processes the time-series data sequentially, where the hidden state vector for the j^{th} day is computed as:

$$h_{ij}^{Sq} = \text{GRU}(V_{ij}^{Sq}, h_{i(j-1)}^{Sq}), \tag{1}$$

where $h_{i(j-1)}^{Sq}$ is the hidden state vector of the $(j-1)^{\text{th}}$ day, and h_{ij}^{Sq} represents the updated state for day j.

Once the daily embeddings are generated, we apply an attention mechanism to dynamically assign importance weights to each day within the week. The attention mechanism computes these weights α_j^{Sq} using a feed-forward neural network. The weighted sum of the daily embeddings is then aggregated to produce a weekly embedding $A(i)^{(Sq)}$, which encapsulates the short-term sequential patterns for stock S_q over week i. The equations are as follows:

$$\alpha_j^{Sq} = \text{Softmax}\left(\tanh(W_0 h_{ij}^{Sq})\right), \tag{2}$$

$$A(i)^{(Sq)} = \sum_j \alpha_j^{Sq} \cdot h_{ij}^{Sq}. \tag{3}$$

Here, W_0 is a learnable parameter matrix in the attention mechanism. This process enables the model to focus on the most significant days within the week and creates a robust representation of the stock's short-term behavior. The weekly embeddings $A(i)^{(Sq)}$ serve as input for subsequent intra-sector relation modeling.

2.3 Intra-Sector Relation Modeling

Input : Embeddings + Adjacency Matrix
Output: Node representations

To model relationships among stocks within the same sector, a fully connected graph is constructed for each sector π_c. The graph $G_{\pi_c} = (M_{\pi_c}, E_{\pi_c})$ consists of nodes M_{π_c}, which represent companies in sector π_c, and edges E_{π_c}, which represent the interactions among these companies. Each node in the graph corresponds to a stock, and its initial feature is the weekly embedding a_i^{Sq}, which encapsulates the short-term sequential patterns for stock S_q during week i.

The relationships among the stocks are modeled using a Graph Attention Network (GAT). The GAT learns attention weights that quantify the influence of neighboring stocks, enabling it to aggregate information from related stocks dynamically.(Figure 4).

For a given stock S_q, the refined embedding g_i^{Sq} is computed by aggregating the features from its neighbors $S_n \in \Gamma(S_q)$ as follows:

$$GAT(G_{\pi_c}; S_q) = g_i^{Sq} = \text{ReLU}\left(\sum_{S_n \in \Gamma(S_q)} \beta_{qn} W_1 a_i^{S_n}\right), \tag{4}$$

where W_1 is a learnable weight matrix, and β_{qn} is the attention weight between stock S_q and its neighbor S_n.

These attention weights are computed using a shared mechanism:

$$\beta_{qn} = \frac{\exp\left(\text{LeakyReLU}(u^T[Wa_i^{S_q} \| Wa_i^{S_n}])\right)}{\sum_{n \in \Gamma(S_q)} \exp\left(\text{LeakyReLU}(u^T[Wa_i^{S_q} \| Wa_i^{S_n}])\right)}, \tag{5}$$

where u is a learnable weight vector, W is a learnable weight matrix, and $\|$ denotes vector concatenation.

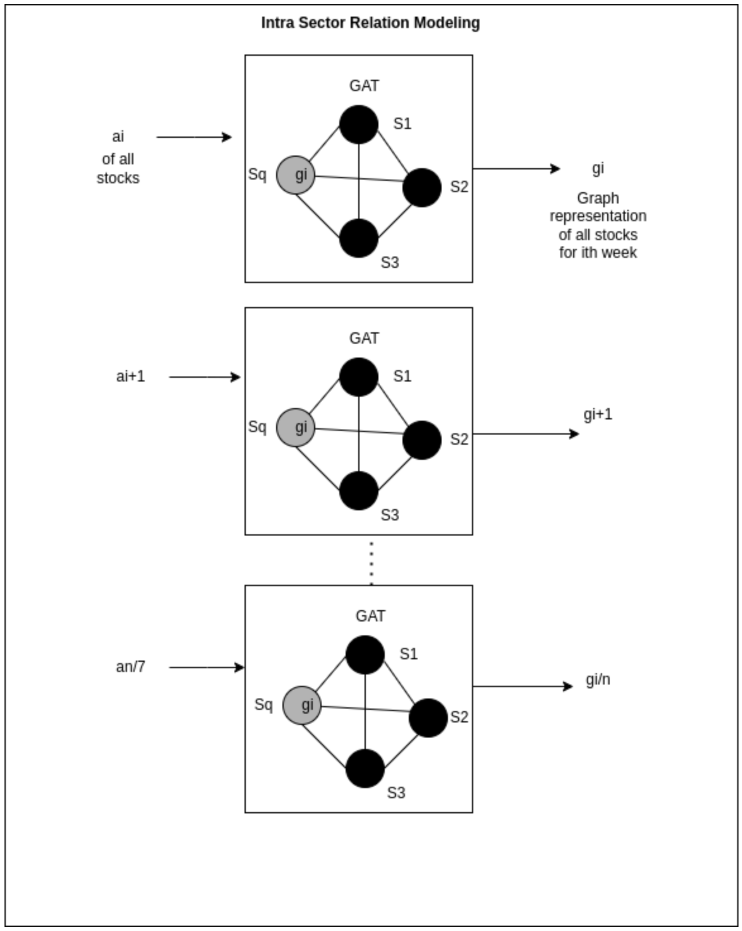

Figure 4: Intra-Sector Relation Modeling with GAT

This process refines the weekly embeddings a_i^{Sq} into graph-enhanced embeddings g_i^{Sq}, which capture both the stock's individual behavior and its relationships with neighboring stocks in the same sector.

For each stock S_q, the weekly embedding a_i^{Sq} is passed through the GAT layer. The GAT aggregates information from neighboring nodes and outputs a refined embedding g_i^{Sq}. This process is repeated for all 500 stocks, generating graph-enhanced embeddings $g_i^{S1}, g_i^{S2}, \ldots, g_i^{S500}$. Each embedding g_i^{Sq} represents the updated state of stock S_q for week i, enriched with intra-sector relational information.

This procedure ensures that the embeddings for each stock are not only informed by its own features but also by the dynamics of other stocks within the same sector. The refined embeddings g_i^{Sq} serve as inputs for subsequent inter-sector modeling or prediction tasks, encapsulating both the stock-specific

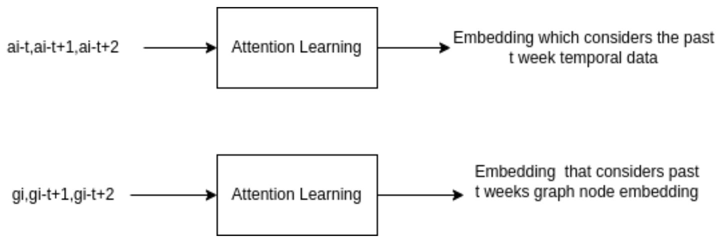

Figure 5: Long term Sequential Learning

and relational insights effectively.

2.4 Long-Term Sequential Learning with Short-Term Embeddings

Input : short-term embeddings g_i^{Sq} and stock-specific embeddings a_i^{Sq}
Output: Sector level trend$\tau_G^i(S_q)$ and stock level trend $\tau_A^i(S_q)$

To effectively capture both short-term and long-term sequential features, the framework considers two distinct sequences of embeddings. The short-term embeddings g_i^{Sq} represent intra-sector relations for stock S_q, while the primitive stock-specific embeddings a_i^{Sq} capture individual stock-level behavior.

Assume that the past t weeks are used to learn long-term features of a stock. Accordingly, we define two sequences of short-term embeddings as:

$$U_G^i(S_q) = \{g_{i-t}^{Sq}, g_{i-(t-1)}^{Sq}, \ldots, g_{i-1}^{Sq}\}, \tag{6}$$
$$U_A^i(S_q) = \{a_{i-t}^{Sq}, a_{i-(t-1)}^{Sq}, \ldots, a_{i-1}^{Sq}\}, \tag{7}$$

where $U_G^i(S_q)$ and $U_A^i(S_q)$ are the embedding sequences from week $i-t$ to week $i-1$, capturing sector-level and stock-level trends, respectively.

To generate long-term embedding vectors, we separately apply an attentive GRU network to the sequences $U_G^i(S_q)$ and $U_A^i(S_q)$. These processes are represented (Figure 5 as:

$$\tau_G^i(S_q) = \text{Attention}\left(U_G^i(S_q)\right), \tag{8}$$
$$\tau_A^i(S_q) = \text{Attention}\left(U_A^i(S_q)\right), \tag{9}$$

where:

- $\tau_G^i(S_q)$: The long-term embedding vector capturing cumulative intra-sector relations up to week i.
- $\tau_A^i(S_q)$: The long-term embedding vector capturing cumulative stock-level trends up to week i.

Attention Mechanism The attention mechanism dynamically assigns importance to embeddings within the look-back period t. For a sequence U, the attention is computed as:

$$\alpha_j = \frac{\exp\left(W^\top \tanh(V u_j)\right)}{\sum_{k=i-t}^{i-1} \exp\left(W^\top \tanh(V u_k)\right)}, \tag{10}$$

where:

- u_j: Embedding for the j^{th} week.
- W and V: Learnable parameters for the attention mechanism.
- α_j: Normalized attention weight for the j^{th} week.

Using the attention weights α_j, the long-term embedding is computed as:

$$\tau = \sum_{j=i-t}^{i-1} \alpha_j u_j. \tag{11}$$

Summary The outputs $\tau_G^i(S_q)$ and $\tau_A^i(S_q)$ serve as the long-term sequential representations for sector-level and stock-level trends, respectively. By incorporating the attentive GRU mechanism, the model effectively balances short-term dynamics and long-term patterns, ensuring robust feature representation for downstream tasks.