Temporal Graph Convolutional Networks for Predicting Stock Movement

Nithya Kemp
Carnegie Mellon University
Pittsburgh, PA 15213
nrkemp@andrew.cmu.edu

Siddharth Parekh
Carnegie Mellon University
Pittsburgh, PA 15213
spparekh@andrew.cmu.edu

Tanay Bennur
Carnegie Mellon University
Pittsburgh, PA 15213
tbennur@andrew.cmu.edu

Abstract

The stock market is notoriously challenging to predict, prompting continuous advancements in forecasting models. A common forecasting task is binary stock movement prediction - predicting whether a stock price will increase or decrease. In this paper, we attempt to increase movement prediction accuracy via a novel ternary movement labeling scheme with separate labels for when a stock price significantly increases, significantly decreases, or remains relatively the same. We introduce a Temporal Graph Convolutional Network (TGCN)-based architecture, trained on five years of historical data from companies in the S&P 500 and a larger superset of 2,415 stocks from the NYSE and NASDAQ Exchanges. Leveraging the temporal capabilities of the TGCN, the approach captures the temporal dependencies inherent in the stock market's complex network, as well as the relationships (edges) between stocks (vertices) both across and within sectors. The proposed model reaches an accuracy of 37.18%, an increase from the common market baseline of random chance at about 33.33%. The results indicate that this architecture merits further research and consideration for financial forecasting, particularly regarding the ternary prediction scheme.

1 Introduction

This paper presents a model designed to forecast stock movements, or whether stock prices will increase or decrease. Stock forecasting is a critical focus for professionals in financial fields, including analysts, traders, and investors. The ability to anticipate how stock prices will change is pivotal, as it often informs key decisions in these professions. Furthermore, stock movements serve as key indicators of company health, attracting interest from a broad range of stakeholders such as policymakers, researchers, and corporate management. Accurate predictions of outcomes have substantial implications for risk management, portfolio optimization, and strategic market planning.

In this study, we employ the Temporal Graph Convolutional Network (TGCN) architecture introduced by Zhao et al. [2020] to analyze and predict stock price movement. By incorporating the temporal dependencies of historical stock data and modeling relationships among companies in the US market, this approach aims to enhance predictive accuracy. The graph-based structure of the TGCN captures sectoral and inter-company interactions, offering a novel way to model the intricate and dynamic relationships that influence stock performance. We train and evaluate our model using stocks from the S&P 500 as well as an overlapping set of 2,415 stocks from the NYSE and NASDAQ Exchanges, with price data from the last five years. The larger dataset is a subset of all equities traded on those exchanged, filtered ensure quality of data and companies based in the US.

Standard approaches to movement prediction use a binary labeling scheme where stocks are labeled as either increasing or decreasing. We extend this scheme into a ternary price prediction scheme - stocks are labeled as either significantly increasing, significantly decreasing, or remaining approximately

Preprint. Under review.

the same. This approach is designed to reduce noise in stock movement predictions, improving overall accuracy and allowing for more performant models. Our approach will contribute not only to the predictive modeling of stock prices but also to the understanding of the temporal and relational dynamics in financial markets. This paper presents a novel stock movement prediction methodology, offering insights into the applicability of TGCNs in financial movement modeling tasks.

2 Background

In our midway report, we detailed our initial objective of predicting stock prices, a task that has been extensively studied within the domain of financial forecasting. To tackle this challenge, we implemented a TGCN designed to predict stock prices over five years using data from companies in the S&P 100 index. Despite data preprocessing efforts, feature selection, and model tuning, the resulting model exhibited significant overfitting. The predictions converged to values closer to zero, diverging substantially from the ground truth and demonstrating poor generalizability on test data.

Recognizing the challenges inherent in predicting exact stock prices—such as high market volatility—we reevaluated our approach. Instead of continuing with an objective that has well-established solutions and benchmarks but is constrained by practical predictability limits, we opted for a more novel and attainable goal within our project's timeframe. Our pivot focused on leveraging a similar TGCN architecture to determine stock movements as opposed to the prices themselves, using a ternary prediction scheme in which a stock's price either significantly increases, significantly decreases, or stays roughly the same (defined by thresholds described later) when compared to the day before. This problem, while adjacent to our original objective, is less researched and thus presents an opportunity for more innovative contributions. We also shifted to larger datasets to counteract overfitting.

This shift not only aligned better with the project's time constraints but also allowed us to explore a domain where our methods could have a higher impact due to the relative scarcity of prior work applying GNN architecture to stock movement problems using this prediction scheme. The redefinition of our objective also required reevaluating the dataset and modifying the architecture to ensure its suitability for trend prediction.

3 Related Work

The field of financial prediction has seen significant advancements through the application of machine learning and deep learning techniques, particularly in stock price prediction. Several groundbreaking approaches have emerged, including the use of Generative Adversarial Networks with Gated Recurrent Units and CNNs for multi-step predictions in a paper by Lin et al. [2021], demonstrating superior accuracy compared to traditional methods. The development of multi-modality graph neural networks (MAGNN) by Cheng et al. [2022] has further enhanced financial time series forecasting by effectively learning from multimodal inputs through heterogeneous graph networks and sophisticated attention mechanisms. Additionally, innovative models like FactorVAE (Duan et al. [2022]) have successfully integrated dynamic factor models with variational autoencoders, offering new perspectives on cross-sectional stock return prediction.

The foundation for many modern financial prediction models can be traced to seminal works in graph neural networks. Introduced by Veličković et al. [2018], Graph Attention Networks (GAT) introduced a groundbreaking architecture that leverages masked self-attentional layers to address the shortcomings of prior graph-based approaches, enabling the model to specify different weights to different nodes in a neighborhood without requiring expensive matrix operations or prior knowledge of the graph structure. Building upon this, Rossi et al. [2020] introduced Temporal Graph Networks (TGN), which established a framework for deep learning on dynamic graphs, introducing a novel approach to capture temporal dependencies and evolving node features through memory modules and graph-based operators. These fundamental advances in graph neural network architecture have significantly influenced the development of financial prediction models.

Comparative studies have extensively evaluated various prediction methodologies, from classical time series and econometric approaches to advanced machine learning and deep learning models. These studies have consistently shown that modern approaches, particularly MARS for machine learning and LSTM for deep learning, outperform traditional methods in stock price prediction tasks (Chatterjee et al. [2021]). As detailed by Wu et al. [2020], the introduction of general graph neural network frameworks for multivariate time series data has further advanced the field by automatically extracting uni-directed relations among variables through graph learning modules and incorporating novel mix-hop propagation techniques.

The application of temporal Graph Neural Networks (GNNs) by Rossi et al. [2020] represents a particularly promising direction in financial time series prediction. Notable developments include the Temporal and Heterogeneous Graph Neural Network (THGNN) approach by Xiang et al. [2022], which excels at learning dynamic relations among binary price movements by generating daily company relation graphs and utilizing transformer encoders. The development of HTGExplainer (Fan et al. [2021]) has further enhanced this field by providing novel methods for explaining heterogeneous temporal GNNs while preserving temporal dependencies and heterogeneity in subgraph generation.

However, a significant gap exists in the current research landscape. Despite extensive research for stock price and movement prediction, no studies have specifically employed a ternary classification scheme (greater than, less than, or approximately equal to previous price). This gap presents a compelling opportunity for future research to combine temporal GNN architectures with more robust stock movement predictions, potentially leading to more nuanced forecasts of price movements and their market impact. Such research could bridge the crucial gap between technical analysis and fundamental market dynamics, offering valuable insights for both academic understanding and practical investment applications. By focusing on a ternary prediction scheme, this approach could provide a more comprehensive view of stock performance relative to its previous state, enabling investors and analysts to make more informed decisions based on the direction and magnitude of potential price changes.

4 Methods

4.1 Initial Objective: Predicting Stock Earnings Trends

Our initial objective was to solve a ternary classification problem, predicting whether stock earnings exceed, meet, or fall short of expectations. This goal presented a feasible challenge and enabled us to explore less-researched territory within financial prediction. However, we encountered a severe lack of open-source data - only able to gather quarterly earnings calls for the last 5 years - insufficient to train a model at the scale we were hoping for. Although we weren't able to pursue this plan, our data collection helped us gather the price history for 2,415 stocks over the last 5-years which became the basis for our larger experiments

4.2 Revised Objective: Predicting Stock Prices with Temporal GNNs

4.2.1 Definitions

We revised this goal to predict a ternary stock movement. We labeled the movement of each stock as either increasing, decreasing or approximately equal (AE), where increasing stocks increased by a significant margin, decreasing stocks decreased by a margin and approximately equal stocks didn't shift enough to be classified as increasing or decreasing.

We defined these margins using a volatility-based threshold, calculated as the product of the stock's standard deviation of percentage change over a given period and tunable hyperparameters α and β. This margin-setting approach distinguished our method from previous literature (e.g., [insert paper name]), and accounts for the varying volatility of different stocks, which is critical for robust stock trend classification.

More rigorously, let $s_i(x)$ be the closing price of the stock i on day t. At time t, the percentage change is defined by,

$$p_i(t_1, t_2) = \frac{s_i(t_2) - s_i(t_1)}{s_i(t_1)} \times 100$$

The thresholds are used to compare the price changes with the volatility of the stock within a rolling window of 20 days, defined as,

$$v_i(t') = \text{stdev}(\{p_i(t-1, t) \mid t \in \{t' - 20, \dots t'\}\})$$

Using these, we obtain the target $\hat{y}_i(t)$ as,

$$\hat{y}_i(t) = \begin{cases} -1 & p_i(t-1,t) < -\alpha v_i(t) \\ 0 & -\alpha v_i(t) \leq p_i(t-1,t) \leq \beta v_i(t) \\ 1 & p_i(t-1,t) > \beta v_i(t) \end{cases}$$

4.2.2 Motivation for Ternary Prediction

The lecture on diffusion models grounded some of our intuition for why ternary prediction might be a more approachable problem than binary prediction with some theoretical basis. Movement of stock prices has been frequently modelled by the assumption of Brownian motion (Mensah et al. [2023], Bongiorno et al. [2017]). We assume that changes in the stock prices are modelled by Ito-diffusion,

$$dx = f(x,t)\,dt + g(t)\,dw$$

where our model is trying to learn f, we observe that as the price change would deviate further from 0 i.e. the price staying the same, the likelihood that the change was derived from f compared to that of being derived from the Brownian motion term g increases since it's defining characteristic is that,

$$w(t+u) - w(t) \sim \mathcal{N}(0, \mu)$$

i.e. g term is 0 in expectation. Our goal with ternary prediction was to essentially be able to determine the tails of this distribution of stock price percentage changes where we become increasingly certain of significant changes (upwards or downwards) in price. By introducing the approximately equal class, we allow the model to default to it in case of outliers (market crash, pandemic, etc.), and use it when it is unsure of the true classification. Therefore the positive and negative predictions are reserved for detecting noticeable positive or negative changes like earnings calls, and overall market trends.

4.3 Model Architecture

For our revised problem, we retained the TGCN architecture due to its ability to process both temporal and relational data effectively. Specifically, we combined a TGCN layer and the Graph Attention Convolution layer introduced by Veličković et al. [2018]. The TGCN layer captures long-term temporal patterns across the graphs. Furthermore, it was used by Xiang et al. [2022] as a baseline for the more popular binary version of the same task. The GATConv layer enhances the model's focus on key nodes and edges, allowing it to adapt to the importance of specific stocks or sectors while harnessing recent developments in the transformer architecture class.

This approach leverages graph representations to model complex relationships between stocks, capturing how financial factors interconnect to influence movements. The temporal aspect enables the model to learn long-term dependencies in data, which are crucial for modelling market dynamics.

4.4 Data and Feature Engineering

Our dataset comprised stock price data for stocks in the S&P 500 and an overlapping set of 2,415 stocks from the NYSE and NASDAQ Exchanges, spanning the past five years. We supplemented this with historical stock price data to calculate the required node features and edge weights. All features were normalized, and missing data was imputed as 0 to ensure consistency.

Taking inspiration from Xiang et al. [2022], we construct dynamic graphs where edge weights at each timestep were based on the correlation of rolling percentage returns between stocks. Edges were established only when the absolute correlation exceeded a predefined threshold, resulting in graphs that naturally partition stocks into traditional financial sectors.

The following features were then computed to represent each stock's corresponding node at each timestep (1 day) t:

- Return: $p_i(t-1,t)$
- Momentum: $p_i(t-20,t)$
- Volatility: $v_i(t)$ as defined earlier.

- Moving Average (20 days): $\frac{1}{20}\sum_{i=1}^{20} s_i(t)$
- Moving Average (50 days): $\frac{1}{50}\sum_{i=1}^{50} s_i(t)$
- Price Movement: $\hat{y}_i(t)$

We thus train the model to predict $\hat{y}(t)$ across timesteps in an attempt to minimise crossentropy loss, an appropriate choice for a multi-classification problem.

5 Results

5.1 Ablations

We ran preliminary experiments with a single TGCN layer in isolation. Despite including dynamic edges and edge weights, this model performed worse than random chance ($\approx 33\%$). In order to strengthen the graph structure, which may have been weakened after repeated passes through the TGCN, we added the GATConv Layer which reliably improved the model's performance. For the sake of brevity and lack of any meaningful results from these initial experiments, we have excluded them from this report.

Once the architecture was finalized, we conducted extensive hyperparameter tuning to refine the model. Key parameters included:

- Correlation threshold for edge construction where edges were only connected if the correlation between the price changes for the last 20 days exceeded this threshold. As described earlier, this would approximately partition stocks into the industrial sectors
- Hyperparameters α and β for defining the volatility margin (which we set to 0.4)
- Learning rate, batch size, epochs, hidden layer size, and the number of layers in the TGCN,
- Number of recurrent steps run through the TGCN i.e. how many timesteps worth of data is being used to predict each new timestep. For r recurrent steps, our model f then predicts as follows,
$$f(\{G_\tau | \tau \in \{t-r, \dots t-1\}\}) = \hat{y}(t)$$
where G_τ represents the graph at time τ.
- Number of attention heads in the GATConv layer.

Table 1 summarizes the results of our ablations, demonstrating the impact of each parameter on model performance.

Due to the ternary nature of the problem, we considered a variety of metrics that have been modified slightly to fit our situation.

- Binary metrics are calculated without considering points that our model predicts as approximately equal.
- Recall and Precision are expanded to the 3 classes by specifically considering each class separately.

Settings										
Recurrence Time	1	1	1	1	1	1	1	1	7	7
Minimum Edge Density	0.3	0.3	0.4	0.35	0.3	0.25	0.2	0.25	0.3	0.4
Heads	4	4	4	4	8	4	4	4	8	8
Hidden Layer Size	64	64	64	64	64	64	64	64	128	64
Epochs	30	30	30	30	30	30	30	30	30	30
Above Bound	0.4	0.3	0.35	0.25	0.3	0.3	0.3	0.25	0.3	0.4
Below Bound	0.4	0.2	0.25	0.2	0.2	0.2	0.2	0.7	0.2	0.4
Results										
Overall Binary Accuracy (%)	49.50	50.57	49.94	52.25	50.41	50.92	49.27	52.71	48.11	49.64
Overall Ternary Accuracy (%)	43.40	31.70	32.09	35.37	31.36	31.51	30.42	44.38	36.84	34.79
Binary Negative Precision (%)	42.72	49.10	48.46	100.00	48.67	49.44	47.35	0.00	44.85	45.78
Binary Positive Precision (%)	50.07	51.62	51.14	51.80	51.32	51.75	50.58	52.71	54.50	57.21
Negative Ternary Recall (%)	2.02	43.85	54.85	0.00	36.25	36.62	43.56	0.00	67.73	36.60
AE Ternary Recall (%)	71.83	0.29	0.29	0.31	0.29	0.29	0.28	48.12	0.41	51.11
Positive Ternary Recall (%)	30.09	56.12	47.66	100.00	61.76	61.90	52.52	58.42	31.92	19.50
Negative Ternary Precision (%)	19.27	30.94	35.78	0.00	31.04	29.64	31.31	0.00	35.98	29.05
AE Ternary Precision (%)	49.26	100.00	100.00	100.00	100.00	100.00	100.00	51.13	75.76	38.95
Positive Ternary Precision (%)	31.30	32.11	28.94	35.63	31.41	32.46	29.69	38.57	38.37	39.99

Table 1: Ablation Log for TGCN/GATConv Architecture on the S&P 500

Due to a lack of time and long training times, we were able to run one experiment on the larger dataset of 2,415 stocks with our best hyperparameters setting the correlation bound to 0.4, $\alpha = \beta = 0.4$, with hidden dimensions set to 64, and 8 attention heads and training the model for 15 epochs. This model ended up being our overall best model surpassing the results of any of our previous experiments.

Best Results					
Overall Bin Acc	**Overall Tern Acc**	**Bin Neg Recall**	**Bin Pos Recall**	**Bin Neg Precision**	**Bin Pos Precision**
53.69%	37.18%	4.15%	1.70%	49.12%	68.41%
Tern Neg Recall	**Tern AE Recall**	**Tern Pos Recall**	**Tern Neg Precision**	**Tern AE Precision**	**Tern Pos Precision**
4.57%	95.46%	1.96%	32.50%	37.19%	51.90%

Table 2: Results for TGCN/GATConv model trained on 2,415 stocks

6 Discussion and Analysis

Though assuming the Markov property and setting the recurrence steps $r = 1$ produced some decent results, they tended to be skewed one way or another either predicting no negative or no positive changes. Setting $r = 7$ produced decent results with some intuition on increasing it a little past the working week to have some insight into the previous week. While it may be a worthwhile exploration in the future to increase this hyperparameter, allowing the model to access larger timesteps like a month or up to a year, due to computational and time constraints, we limited ourselves to using $r = 7$. Though our best results for the S&P 500 dataset were produced by a correlation bound of 0.3, we decided to go with a bound of 0.4 for training the model on the larger dataset to make the graph slightly sparser to counteract the four-fold increase in information.

Our best model does indeed achieve our goal to a certain extent. With an overall accuracy on the ternary task of 37.18%, the model performs better than random chance, a common bound for market prediction due to the efficient market hypothesis. Similar results (close to random chance) have been seen in Xiang et al. [2022] where their model achieved 58% accuracy on binary prediction. However, this slight edge is what financial investment institutions operate on - using a marginal expected benefit repeatedly to achieve consistent profits. Xiang et al. [2022] also evaluated a baseline TGCN architecture for binary prediction obtaining an accuracy of 53%. While our binary accuracy is very similar, it cannot be interpreted the same way since the metric excludes all approximately equal predictions which form the bulk of our outputs.

The model primarily tends to predict to the approximately equal class resulting in our recall being rather low. However, we obtain good precision values for binary positive and negative prediction i.e. when the model predicts a positive change, 68.41% of the time it is indeed a positive change. Similarly, when the model predicts a negative change, 49.12% of the time, it is indeed a negative change. Particularly for the positive case, such a model could be incorporated into an ensemble of models to strengthen the confidence in a particular stock gaining value. Figure 1 demonstrates this phenomenon where the model's positive and negative predictions have distributions that are skewed in those directions respectively. Note that the y-axis has been normalised to the same scale for both the histograms.

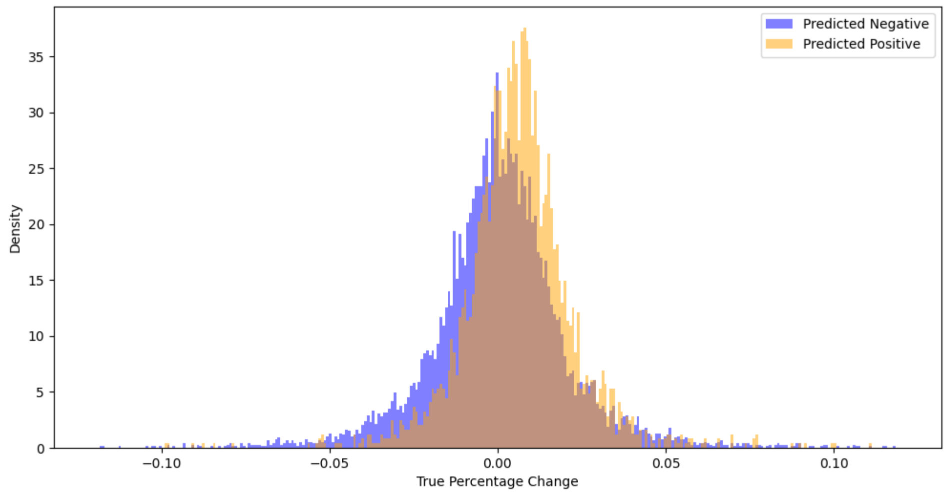

Figure 1: Distribution of Model's Binary Predictions

Despite this phenomenon which could be exploited to generate returns in the market, the ternary accuracy and low recall suggest significant room for improvement. The TGCN architecture has proven itself effective in modelling temporal dependencies in graphs Zhao et al. [2020] and Graph Neural Network (GNN)-based models have, in general, demonstrated their applicability in finance. However, we realise that this architecture may not be particularly suited to solve the ternary version of the problem. Figure 2 shows the distribution of the ground truth values of the points predicted to be approximately equal. Though being centered very close to 0.015 (the average daily change of any stock in our dataset over time), it goes past the thresholds set by the volatility, dominating the predictions of the model.

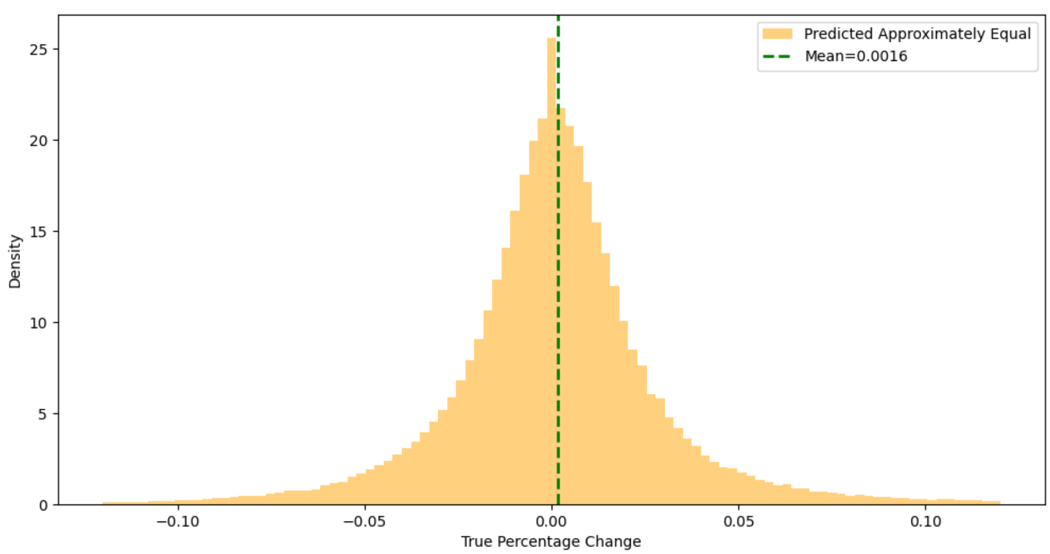

Figure 2: Distribution of Approximately Equal Predictions

Though our initial intuition suggested this might be an easier problem than binary movement prediction or real-valued price prediction, through a deeper analysis of our results and a better understanding of the nature of each of these problems, we see that this ternary movement prediction task with variable thresholds might be as difficult as real-valued price prediction. Knowing the volatility for the past timesteps, the real-valued price prediction problem could be reduced to the ternary movement prediction by setting specific thresholds that would enable us to predict whether the price movement (upward or downward) would be beyond a given price, i.e. we would be able to predict how much the stock price will change through trial-and-error with the thresholds.

7 Conclusion

This exploration of ternary stock movement prediction using Temporal Graph Convolutional Networks has revealed important insights into the complexity of financial forecasting. While the model achieved an overall accuracy of 37.18% on the ternary task and demonstrated promising precision rates of 68.41% for positive predictions, these results highlight both the potential and limitations of the current approach . The ternary prediction problem proved to be more challenging than initially anticipated. Rather than simplifying the binary classification task, the introduction of an "approximately equal" class added complexity to the prediction landscape. This is evidenced by the model's tendency to default to the approximately equal class, resulting in low recall values for positive and negative predictions. Possible future directions include but are not limited to enhancing the architectures by exploring longer temporal dependencies beyond the current 7-day recurrence window, alternative graph construction methods which might better capture the complex relationships between stock prices, and finally further integration with other types of financial data, such as earnings reports and market sentiment which could provide additional predictive power.

8 Teammates and Work Divison

This team is comprised of three undergraduate students at Carnegie Mellon University: Nithya Kemp (B.S. Artificial Intelligence), Siddharth Parekh (B.S. Computer Science), Tanay Bennur (B.S. Computer Science & Artificial Intelligence).

All teammates had involvement with every aspect of this paper and contributed equally. Table 3 illustrates areas in which each teammate took special initiative and responsibility to lead and accomplish.

Table 3: Teammates and Work Division

Teammate	**Tasks**
Nithya (nrkemp)	· Evaluation and analysis: Define metrics (e.g., accuracy, precision, recall) and assess the model's performance. · Visualization and reporting: Create visualizations of data, graph relationships, and results. · Write-up and presentation: Organise sections of the paper, focusing on findings, discussion, and conclusion.
Siddharth (spparekh)	· Data collection and preprocessing: Gather historical stock price data from and clean it for analysis. · Implement temporal feature engineering: Design features that capture temporal dependencies. · Evaluate baseline models: Test simpler models (e.g., logistic regression, random forests) for comparison with the TGCN model.
Tanay (tbennur)	· Graph construction: Define the relationships (edges) between stocks (vertices) using sectoral, correlation, or dependency data. · Develop and implement the TGCN architecture: Code the model using a deep learning framework (e.g., PyTorch or TensorFlow). · Hyperparameter tuning: Optimize the architecture for best performance.

9 Access to Code

Access to our code can be found **here** or at the link below:

https://github.com/nithyuk/TCGNN.git

References

Enea G. Bongiorno, Aldo Goia, and Philippe Vieu. On the geometric brownian motion assumption for financial time series. In Germán Aneiros, Enea G. Bongiorno, Ricardo Cao, and Philippe Vieu, editors, *Functional Statistics and Related Fields*, pages 59–65, Cham, 2017. Springer International Publishing. ISBN 978-3-319-55846-2.

Ananda Chatterjee, Hrisav Bhowmick, and Jaydip Sen. Stock price prediction using time series, econometric, machine learning, and deep learning models. *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, pages 289–296, 2021.

Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 2022.

Y. Duan, L. Wang, Q. Zhang, and J. Li. Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4):4468–4476, 2022.

Yujie Fan, Mingxuan Ju, Chuxu Zhang, Liang Zhao, and Yanfang Ye. Heterogeneous temporal graph neural network, 2021. URL `https://arxiv.org/abs/2110.13889`.

Hung-Yang Lin, Chen Chen, Gaofeng Huang, and Amir Jafari. Stock price prediction using generative adversarial networks. *Journal of Computer Science*, 2021.

Eric Teye Mensah, Alexander Boateng, Nana Kena Frempong, and Daniel Maposa. Simulating stock prices using geometric brownian motion model under normal and convoluted distributional assumptions. *Scientific African*, 19:e01556, 2023. ISSN 2468-2276. doi: https://doi.org/10.1016/j.sciaf.2023.e01556. URL `https://www.sciencedirect.com/science/article/pii/S2468227623000157`.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs, 2020. URL `https://arxiv.org/abs/2006.10637`.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL `https://arxiv.org/abs/1710.10903`.

Zonghan Wu et al. Connecting the dots: Multivariate time series forecasting with graph neural networks. *arXiv preprint arXiv:2005.11650*, 2020.

Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. Temporal and heterogeneous graph neural network for financial time series prediction. In *Proceedings of the 31st ACM International Conference on Information amp; Knowledge Management*, CIKM ’22. ACM, October 2022. doi: 10.1145/3511808.3557089. URL `http://dx.doi.org/10.1145/3511808.3557089`.

Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, September 2020. ISSN 1558-0016. doi: 10.1109/tits.2019.2935152. URL `http://dx.doi.org/10.1109/TITS.2019.2935152`.