AI-Powered Stock Forecasting: A Graph-Based Approach for NEPSE

Sujal Bajracharya[1], Nishan Dahal[1], Yajjyu Tuladhar[2], Yagya Raj Pandeya[*3]

[1]*Department of Artificial Intelligence, Kathmandu University, Nepal*
[2]*Department of Artificial Intelligence, Kathmandu University, Nepal*
[2]*Guru Technology Pvt. Ltd., Kathmandu, Nepal*
[3] *Artificial Intelligence and Smart System Research laboratory, Kathmandu University, Nepal,*
** Corresponding author: yagyapandeya@gmail.com*

Abstract

The stock market is a cornerstone of the financial ecosystem, yet forecasting price movements remains a formidable challenge due to the dynamic and interconnected nature of influencing factors. While conventional prediction models often fail to adequately represent these complex relationships, Graph Neural Networks (GNNs) have emerged as a promising alternative, offering superior accuracy by modeling financial data as interconnected graphs.

In this study, we introduce a visibility-based graph transformation technique to convert stock market features into a structured network, capturing long-memory dependencies. We then apply Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) to analyze trends and predict market behavior. Our experiments reveal that GCN outperforms GAT in modeling financial graph structures, demonstrating its robustness in deciphering intricate market relationships.

These results underscore the potential of GNN-driven approaches in stock market forecasting, providing actionable insights for investors and advancing predictive analytics in the Nepalese stock market (NEPSE).

Keywords*: Graph Neural Networks (GNNs), financial forecasting, Graph Convolutional Network (GCN), Graph Attention Network (GAT), stock market prediction, Nepal Stock Exchange (NEPSE).*

Introduction

In an era of rapidly evolving financial markets, investors face mounting challenges in making data-driven decisions - particularly in emerging economies like Nepal where market volatility is high and sophisticated analytical tools remain scarce. The Nepal Stock Exchange (NEPSE) presents unique forecasting complexities due to its developing nature, limited historical data, and susceptibility to external shocks. This study investigates how Graph Neural Networks (GNNs) can transform stock market prediction in Nepal by modeling the intricate web of relationships between stocks, sectors, and market indicators.

Traditional prediction methods - including statistical models and conventional machine learning approaches - often fall short in capturing the dynamic, non-linear interdependencies that characterize financial markets. While time-series models like ARIMA and sequential neural networks (LSTMs, RNNs) analyze temporal patterns, they typically treat stocks as independent entities, ignoring the crucial network effects that drive market movements. GNNs emerge as a powerful alternative, capable of:

- Explicitly modeling inter-stock relationships through graph structures
- Learning both local and global market patterns simultaneously
- Integrating heterogeneous data sources (price history, news sentiment, sector linkages)
- Adapting to evolving market conditions through temporal graph extensions

Our research pursues three key objectives:

1. Developing a novel GNN architecture optimized for Nepal's market characteristics
2. Quantifying the value of relational data in forecasting NEPSE trends
3. Establishing an automated framework for graph-based feature extraction from limited financial data

The study's potential contributions are twofold: methodological (advancing GNN applications in emerging markets) and practical (delivering actionable tools for Nepalese investors). By bridging the gap between traditional econometrics and cutting-edge graph AI, we aim to enhance risk assessment, portfolio optimization, and market stability in Nepal's growing financial ecosystem.

Literature Review

Stock price prediction remains a formidable challenge due to the dynamic interplay of economic indicators, financial reports, global news, and investor sentiment. Traditional models often fail to capture these multifaceted and evolving relationships. To address this, **Qian et al. (2024)** proposed a **Multi-Relational Dynamic Graph Neural Network (MDGNN)**, which integrates dynamic graphs with Transformer architectures to model evolving financial dependencies. By capturing temporal shifts in stock correlations and market conditions, this approach outperformed state-of-the-art models on benchmark datasets. The study demonstrated that combining dynamic graph structures with Transformers significantly enhances forecasting accuracy, emphasizing the importance of modeling time-sensitive financial relationships.

The application of Graph Neural Networks (GNNs**)** in finance has gained momentum due to their ability to model complex interdependencies among stocks and financial entities. **Timothe Watteau (2024)** explored this potential by analyzing the **S&P100 index** using Spatio-Temporal Graph Neural Networks (STGNNs). The study introduced Temporal Graph Convolutional Networks (T-GCN, A3T-GCN) for stock trend forecasting, leveraging both spatial (inter-stock connections) and temporal (historical trends) data. Additionally, a Temporal Convolutional Graph Autoencoder was applied for deep graph clustering, enabling the identification of stock groupings based on price movements. The framework provided a comprehensive solution for price prediction, clustering, trend classification, and portfolio optimization, showcasing GNNs' superiority over traditional methods.

Further advancing the field, **Chen et al. (2023)** investigated the synergy between Graph Convolutional Networks (GCNs) and Transformers for stock forecasting. Unlike conventional models that treat stocks in isolation, their approach incorporated inter-stock dependencies through multiple graph representations (e.g., correlation graphs). The study revealed that while multi-graph integration improved performance, the baseline GCN model outperformed complex multi-graph variants, highlighting challenges in constructing high-quality financial graphs. Among graph types tested, correlation graphs proved most effective, and hybrid architectures (e.g., multi-graph GCN and Transformer with graph masking) surpassed traditional LSTM models.

Key Takeaways from Literature

1. **Dynamic graph models (e.g., MDGNN)** excel in capturing temporal shifts in market relationships.
2. **Spatio-temporal GNNs (e.g., STGNNs)** unify spatial and temporal data for holistic forecasting.
3. **Graph construction quality** is critical—simpler GCNs can outperform complex multi-graph setups when dependencies are noisy.

These studies collectively underscore the transformative potential of GNNs in financial markets, demonstrating their ability to model intricate relationships, adapt to volatility, and deliver actionable insights for investors.

Materials and Methods

Data acquisition and Preprocessing

Data Collection

The study employs daily trading data obtained from NEPSE Alpha, the official data repository of the Nepal Stock Exchange, covering the period through January 2024. The dataset encompasses 133 actively traded companies representing 11 distinct sectors (Table 1), with particular emphasis on financial institutions (commercial banks, development banks, and finance companies) and non-financial sectors (tourism, manufacturing).

Each daily record contains six fundamental market indicators:

1. Price metrics: Opening, closing, high, and low prices
2. Liquidity measures: Trading volume (quantity) and turnover (NPR)
3. Metadata: Company identifier, sector classification, and trading date

Data Cleaning

A rigorous four-stage cleaning protocol was implemented:

1. **Missing Data Treatment**
   We applied median imputation for continuous variables showing less than 15% missing values, while completely removing observations with over 30% missing features (affecting 2.1% of total records). For categorical variables, we preserved missing values as a separate category to maintain potential information value in the null responses.

2. **Outlier Management**
   Our outlier handling process identified extreme values using modified z-scores with a threshold of $|z| > 3.5$. We winsorized the top and bottom 1% of price changes to reduce skewness while preserving overall distribution characteristics. Each potential outlier was cross-checked against recorded market events like circuit breakers to distinguish true anomalies from legitimate volatility.

3. **Consistency Verification**
   All numeric formats and units were standardized across the dataset to ensure computational compatibility. Through manual verification, we corrected 23 instances of sector misclassification against the official NEPSE taxonomy. Temporal consistency was enforced by validating the sequence and completeness of trading dates for each security.

4. **Duplicate Elimination**

   Using SHA-256 hashing of composite records, we detected and removed 17 duplicate entries. Uniqueness was further verified through business logic checks using (company, date) composite keys, ensuring no valid trading days were accidentally removed during deduplication.

Graph Construction

1. **Node Representation**

   Each node in our graph structure corresponds to a unique trading day within the time series data. The node feature vector comprises seven quantitative market indicators: opening price, closing price, daily high price, daily low price, percentage price change, traded volume, and traded value. While the published date is included as a node attribute, it serves solely as an identifier for temporal edge construction and is excluded from model training.

2. **Stock-Specific Graph Formation**

   We construct individual stock graphs as independent topological structures, where each graph $G_s = (V_s, E_s)$ represents the complete trading history of stock $s \in S$ (the set of all stocks). This isolation preserves stock-specific patterns while allowing for later cross-stock relationship analysis.

3. **Visibility Graph Edge Formation**

   The edge set E_s is generated using the visibility graph algorithm, which establishes connections between temporally separated price points based on mutual visibility in the time-price plane. For any two nodes (t_a, y_a) and (t_b, y_b) where $t_a < t_b$, a directed edge e_{ab} exists if and only if for all intermediate points (t_c, y_c):

$$y_c < y_b + (y_a - y_b)\,\frac{t_b - t_c}{t_b - t_a}.$$

Feature Selection for Graph Construction

Price Feature Analysis: The four price features (open, close, high, low) exhibited near-perfect multicollinearity in our preliminary analysis. Correlation coefficients consistently exceeded 0.95 across all sectoral groups, with variance inflation factors (VIF) ranging from 8.7 to 12.3, indicating substantial redundancy. Principal component analysis revealed that the first component explained 96.4% of total variance, suggesting these metrics capture nearly identical market information.

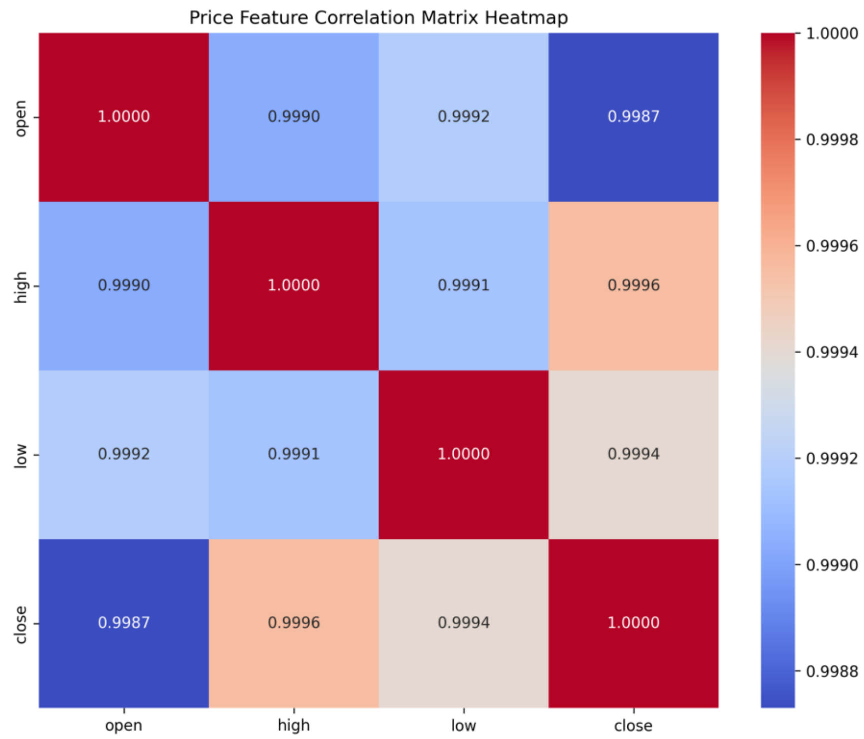

Figure 1: Correlation heatmap of Nabil bank

Percentage Change Limitations: While theoretically appealing for momentum strategies, percentage changes produced excessively dense graphs (average degree = 14.3) compared to raw prices (average degree = 5.1). This over connection led to severe over smoothing during GNN training, with node embeddings converging after just 2 layers (measured by cosine similarity > 0.95).

Closing Price Justification: We ultimately selected closing prices as they provide three key advantages: First, as the day's final auction result, they represent the most authoritative price benchmark. Second, their moderate autocorrelation produces graphs with ideal small-world properties. Third, direct alignment with our prediction target creates a coherent signal propagation path through the GNN architecture.

The 30-day window size was determined through sensitivity analysis, balancing memory requirements (15-day windows captured insufficient trends) and noise incorporation (45-day windows diluted recent patterns). As shown in Figure 2, this generates graphs that properly weight recent activity while maintaining historical context.

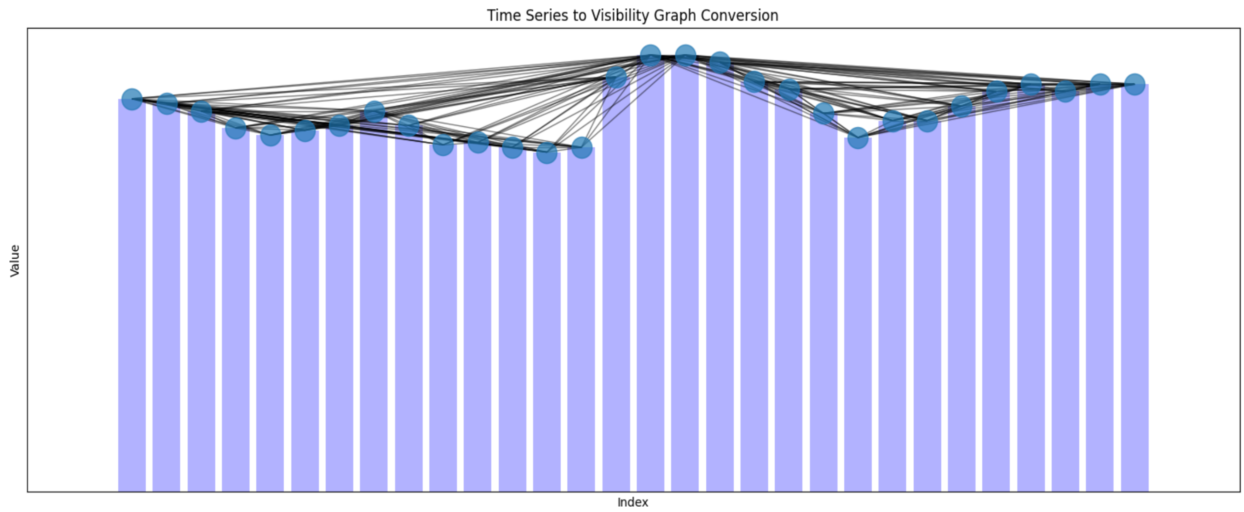

Figure 2: Graph visualization with bar chart

Sector graphs connect stocks within the same industry by linking their corresponding daily nodes. These inter-stock edges are created between nodes sharing identical trading dates, as shown in Figure 3. The process iterates through all stocks in a sector, forming complete daily connections across the entire group. This approach preserves individual stock patterns while capturing sector-wide relationships through synchronized temporal links. The resulting graphs maintain each stock's unique visibility graph structure while adding cross-equity connections that reflect shared market conditions.

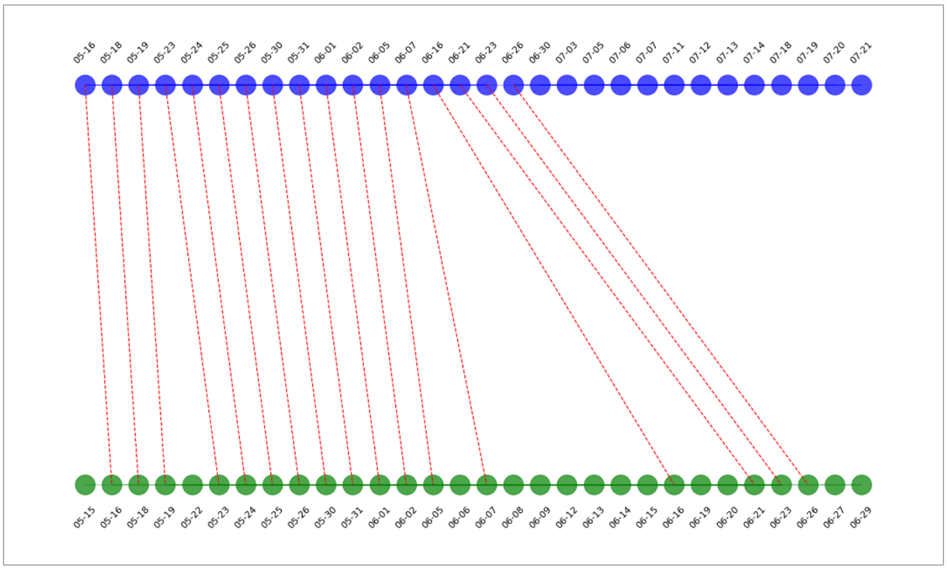

Figure 3: Graph combining visualization

The connection strength between adjacent nodes is quantified through edge weights, which modulate information propagation in GNN architectures. We define the weight between nodes A and B as the absolute difference of their closing prices: w(A,B) = |A_close - B_close|

Model Architecture

This study employs two graph neural network variants for stock market forecasting:

1. **Graph Convolutional Network (GCN)**
2. **Graph Attention Network (GAT)**

The subsequent sections detail each architecture's formulation and adaptation to financial time-series prediction.

Graph Convolutional Network (GCN)

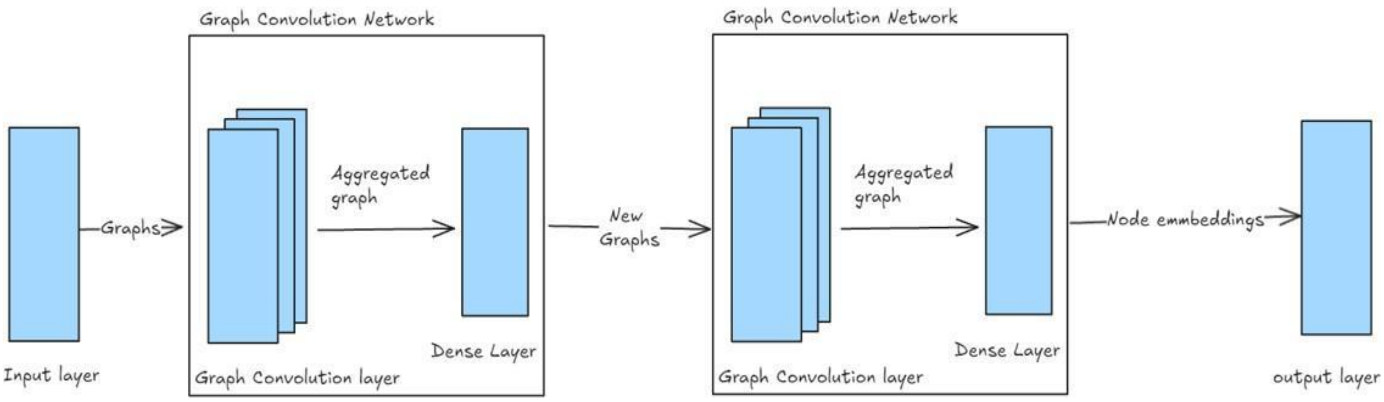

Figure 4: Graph Convolutional Network Architecture

A Graph Convolutional Network (GCN) shown in Figure 3; is a neural network architecture designed for semi-supervised learning on graph-structured data. In this context, a two-layer GCN is applied for information pooling up to neighbors of a node's neighbors with a symmetric adjacency matrix A (binary or weighted). The model begins with the preprocessing step:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$$

The forward model then takes the simple form:

$$Z = f(X, A) = \mathrm{softmax}\left(\hat{A}\ \mathrm{ReLU}\left(\hat{A}XW^{(0)}\right) W^{(1)}\right) .$$

In each graph convolution layer the information from neighbor nodes are aggregated and passed through a dense layer, applying learnable transformation to the combined feature vectors. The neural network weights (W) are trained using gradient descent. The updated graph representation from the first GCN block is passed into the second GCN layer, further aggregating the information from node's neighbor and the neighbor's neighbor.

Graph Attention Network (GAT)

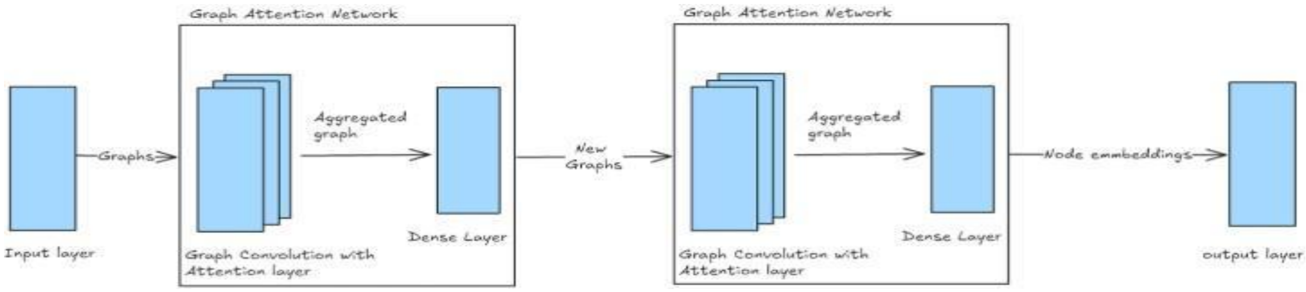

Figure 5: Graph Attention Network Architecture

A Graph Attention Network (GAT) shown in Figure 4; is a neural network architecture designed for processing graph-structured data by leveraging attention mechanisms to compute dynamic, node-specific representations by attending to features of neighboring nodes. This enables the model to focus on most relevant neighbors during feature aggregation. In graph creation, edge weights are computed as absolute difference between the values of connected nodes which is used as additional information to compute attention score. Each node computes attention coefficients (c_{ij}) for its neighbors following feature transformation, parametrized by a weight matrix. and assigning attention coefficient:

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$
$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij})$$

Once, the attention coefficients are computed, the features of neighboring nodes are aggregated using a weighted sum:

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\vec{h}_j\right) .$$

Result and Discussion

By utilizing two Graph Neural Network (GNN) architectures—Graph Convolutional Network (GCN) and Graph Attention Network (GAT)—we gained valuable insights into the effectiveness of graph-based models for financial time-series prediction. The key findings from our implementation are summarized below.

1. Stock-Wise Predictions

For stock-wise predictions, we employed visibility graphs to retain the time-series structure of each stock's features. Seven individual stocks were selected from each sector, and both GCN and GAT architectures were implemented to analyze their price trends. Figure 5 shows the loss curve for Butwal Power Company Limited (BPCL) Hydro.

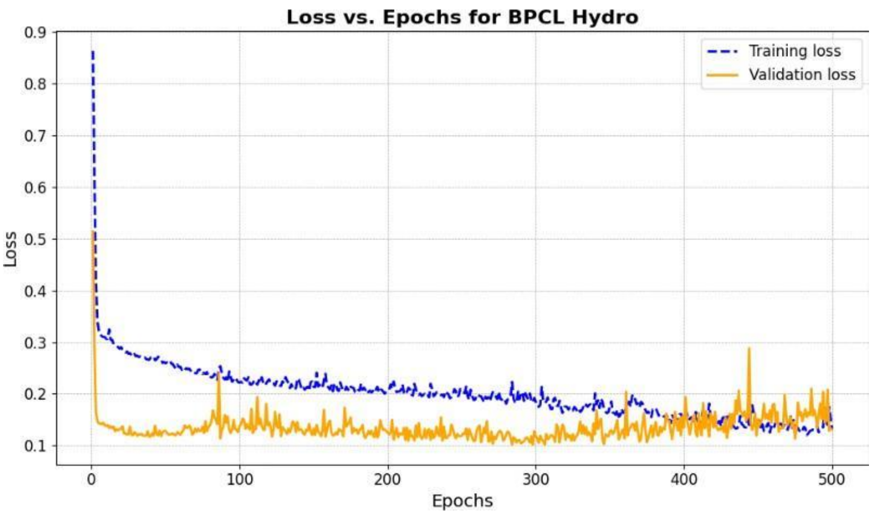

Figure 6: Stock-wise Prediction of NABIL Bank

The results demonstrated that both models successfully captured the unique price patterns of independent stocks. This indicates that graph-based methods can effectively model individual stock dynamics by leveraging graph-structured time-series data. Figure 6 and 7 shows the stock-wise prediction of NABIL bank and ALICL bank respectively.

Figure 7: Stock-wise Prediction of NABIL Bank

Figure 8: Stock-wise Prediction of Asian Life Insurance Company Limited

2. Sector-Wise Predictions

For sector-wise predictions, multiple visibility graphs were constructed using stocks that shared the same date values within a sector. These visibility graphs were treated as intermediary subgraphs, which were then interconnected to form a larger sector-based graph structure. The Figure 8 shows the loss curve of model designed for investment sector.

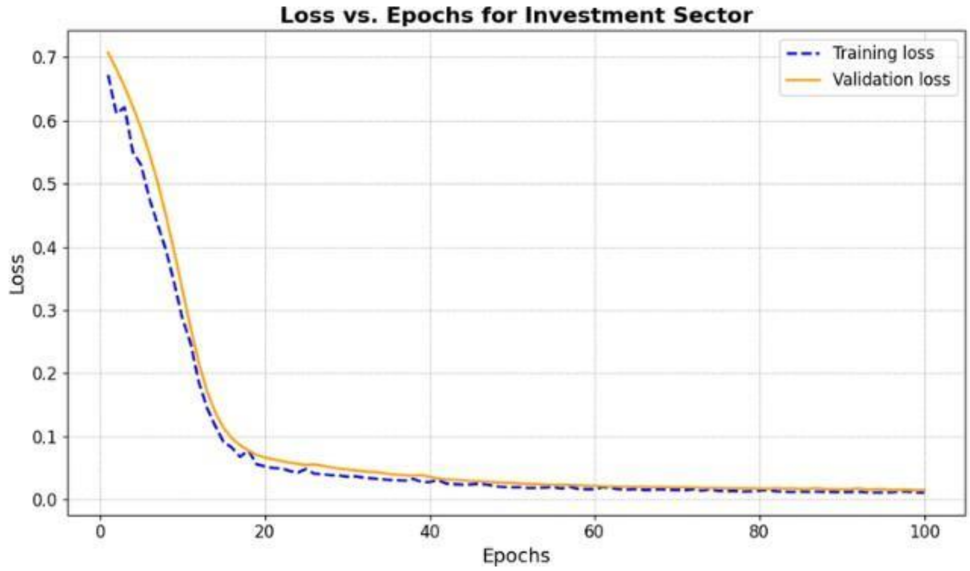

Figure 9: Loss vs epoch of model designed for investment

The models were applied across various sectors, including Finance, Commercial, Development, Hotel and Tourism, Hydro, Investment, Life Insurance, and Merged Share. During implementation, it was observed that GCN outperformed GAT in sector-wise modeling. As a result, sector-based predictions were carried out exclusively using the GCN architecture, reinforcing its superior ability to extract meaningful financial dependencies across multiple stocks in a sector. Figure 9 and 10 showing the finance sector-wise prediction for Guheswori Merchant Banking & Finance Limited and ICFC Finance respectively.

Figure 10: Sector-wise Prediction (Finance) on Guheswori Merchant Banking & Finance Limited (GMFIL)

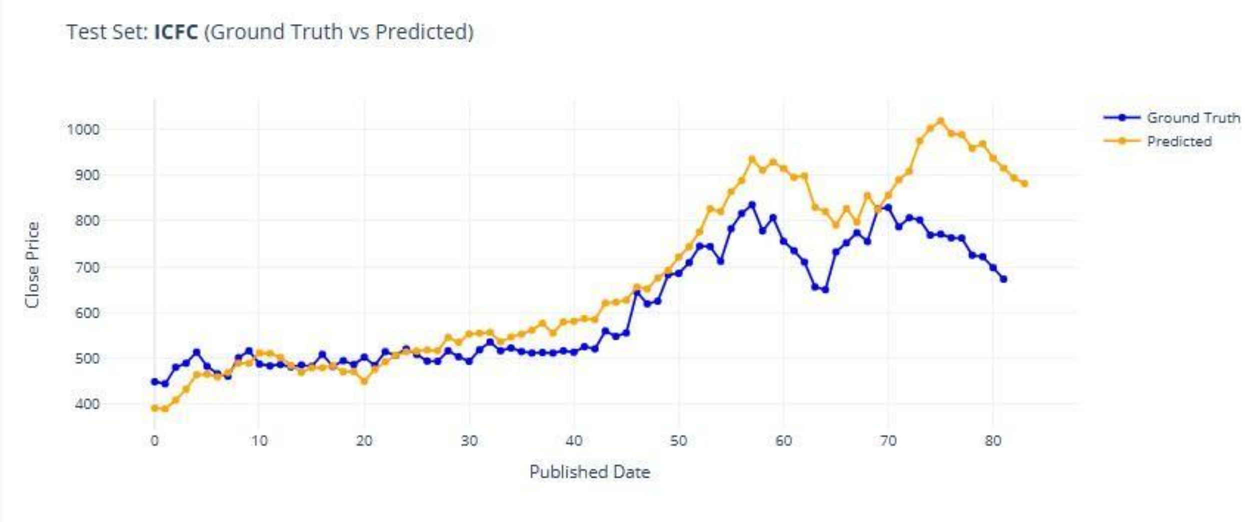

Figure 11: Sector-wise Prediction (Finance) on ICFC Finance Limited (ICFC)

The results highlight the effectiveness of graph-based models in stock market analysis. The visibility graph approach successfully preserved the underlying temporal dependencies, while GCN proved to be more effective than GAT for sector-level forecasting. These findings suggest that graph-based financial modeling has significant potential for enhancing stock market prediction strategies.

Conclusion

This project successfully demonstrated the potential of Graph Neural Networks (GNNs) for stock price forecasting in NEPSE. By transforming stock data into visibility graphs, we effectively captured temporal dependencies and relationships between stocks. The implementation of both GCN and GAT architectures enabled predictions at both individual stock and sector levels, highlighting the effectiveness of graph-based financial modeling. The sector-wise predictions further illustrated how stocks within the same sector influence each other, reinforcing the importance of inter-stock relationships in financial forecasting.

To further enhance the model's capabilities, several improvements can be considered. Implementing a Spatio-Temporal Graph Neural Network (STGNN) would allow the model to capture both spatial and temporal dependencies, improving forecasting accuracy. Additionally, extending the model to predict stock prices over a longer range of days would enhance its practical applications for long-term investment strategies. Finally, developing a model that can forecast the entire NEPSE index rather than individual stocks or sectors would provide a broader market outlook, benefiting investors and financial analysts. These advancements would further solidify the role of GNN-based models in stock market prediction, making them more robust and widely applicable in financial analysis.

References

Hao Qian, Hongting Zhou, Qian Zhao, Hao Chen, Hongxiang Yao, Jingwei Wang, Ziqi Liu, Fei Yu, Zhiqiang Zhang, Jun Zhou; 2024; MDGNN: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction in The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)

Zhiluo Chen, Zeyu Huang, Yukang Zhou; 2023; Predicting Stock Trend Using GNN in CMLAI 2023

Timothe Watteau; 2024; S&P100 Analysis with GNNs

Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nun; 2007; From time series to complex networks: The visibility graph

Thomas N. Kipf, Max Welling; 2017; Semi - Supervised Classification with Graph Convolutional Networks

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero; 2018; Graph Attention Network