# STOCK FORECASTING USING NEURAL NETWORK WITH GRAPHS

**A PROJECT REPORT**

***Submitted by***

**Bhavani Kurimilla – B182629**

**Tilak Thummanapally – B181382**

**Vignesh Embari – B182598**

**Of**

**Bachelor of Technology**

***Under the guidance of***

**P. Laxmi Narayana**

**Assistant Professor**

**DEPARTMENT OF COMPUTER SCIENCEAND ENGINEERING RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIESBASAR, NIRMAL (DIST.), TELANGANA – 504107**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,BASAR**

# CERTIFICATE

This is to certify that the Project Report entitled **'Stock Forecasting using Neural Network with Graphs'** submitted by **Bhavani Kurimilla – B182629, Tilak Thummanapally – B181382, Vignesh Embari – B182598**, Department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, Basar; for partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering; is a bonafide record of the work and investigations carried out by him/her/them under my supervision and guidance.

| | |
|---|---|
| **PROJECT SUPERVISOR** | **HEAD OF DEPARTMENT** |
| **MR. P. LAXMI NARAYANA** | **MR. REVYA NAIK** |
| **ASSISTANT PROFESSOR** | **ASSISTANT PROFESSOR** |

**EXTERNAL EXAMINAR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,BASAR**

## DECLARATION

I/We hereby declare that the work which is being presented in this project entitled, **"STOCK FORECASTING USING NEURAL NETWORK WITH GRAPHS"** submitted to **RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, BASAR** in the partial fulfilment of the requirements for the award of thedegree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING,** is an authentic record of my/our own work carried out under the supervision of **"MR. P. LAXMI NARAYANA", Assistant Professor in Department of Computer Science and Engineering, RGUKT, Basar.**

The matter embodied in this project report has not been submitted by me/us for the award of anyother degree.

Place: RGUKT, BASAR

Date: 06/01/2024

BHAVANI KURIMILLA – B182629

TILAK THUMMANAPALLY - B181382

VIGNESH EMBARI – B182598

# ACKNOWLEDGEMENT

First, we would like to thank management of RGUKT-Basar for giving us the opportunity to do major project within the organization. We also would like to thank all the people who worked along with us in RGUKT-Basar, with their patience and openness they created an enjoyable working environment.

We would like to thank our project guide **P. Laxmi Narayana, Assistant Professor, RGUKT, Basar** for his guidance and help throughout the development of this project work. Without his guidance, cooperation and encouragement, we couldn’t learn many new things during our project tenure.

We would like to thank our Head of the Department, **Mr. Revya Naik, Assistant Professor, RGUKT, Basar** for his support and encouragement in completing our project.

We would like to thank our department project coordinator **Mrs. B. Latha, Assistant Professor, RGUKT, Basar** for their support and advices to get and complete our project within the time.

We are extremely grateful to our department staff members, family members and friends who helped us in successful completion of this major project.

| **Students Names** | **Signatures:** |
|---|---|
| **Bhavani Kurimilla** | |
| **Tilak Thummanapally** | |
| **Vignesh Embari** | |

# ABSTRACT


In the realm of predicting stock prices, dealing with the complexities involved remains a challenge and a topic of interest. The use of deep learning, especially neural network models, has gained popularity in this domain. Current studies often focus on how a stock's historical information influences its future price, but it's crucial to consider the impact of other related stocks.

To address this, we propose integrating stock graphs into a neural network model. We use graphs because their connected structure can capture the relationships between stocks effectively. Our model extends the Graph Convolutional Network (GCN) to handle multiple graph features, going beyond the limitations of single-graph-focused models. We also employ a transformer-based model to understand correlations between stocks, leveraging its effectiveness in natural language processing. In our approach, the stock graph acts as a mask in the attention layer, giving the transformer prior knowledge.

Our experiments use Microsoft Corporation stock data and show that our graph-based model performs better than traditional recurrent neural network models and those not considering graph structures. We explore different graph types and find that combining multiple graphs significantly improves accuracy. However, our model doesn't outperform the general GCN model due to the quality of our constructed graphs. We also introduce three graph construction methods, with results favouring the correlation graph as the optimal choice.

Comparative analyses demonstrate that both multi-graph GCN and transformer with a graph mask outperform the LSTM model. Additionally, a pure transformer combined with LSTM performs better than the standalone LSTM model, supporting our belief that internal relations play a crucial role in improving stock prediction outcomes.

# CONTENTS

**IMPLEMENTATION**



**CONCLUSION**

## LIST OF FIGURES

## LIST OF ABBREVATIONS AND SYMBOLS USED

| S. No | Abbreviation | Expansion |
|---|---|---|
| 1 | RNN | Recurrent Neural Network |
| 2 | CNN | Convolution Neural Network |
| 3 | LSTM | Long Short Term Memory |
| 4 | GFNN | Genetic Fuzzy Neural Networks |
| 5 | SOFNN | Self-Organizing Fuzzy Neural Networks |
| 6 | EMF | Efficient Market Hypothesis |
| 7 | GCN | Graph Convolutional Neural Networks |
| 8 | GNN | Graph neural networks |
| 9 | MAE | Mean Absolute Error |
| 10 | RMSE | Root Mean Squared Error |

# 1. INTRODUCTION

## 1.1 BACKGROUND

Forecasting stock prices is a perennial challenge due to inherent instability and the intricate nature of the market. The stock market is influenced by various factors, making it challenging to assess specific circumstances accurately. For instance, the impact of a newly announced policy on particular sectors is difficult to quantify precisely. Such policy-driven effects on primary industries also resonate in the stock prices of secondary and tertiary sectors. Some earlier studies asserted that stock price movements are random, and stock performance is unpredictable. The Efficient Market Hypothesis (EMH) proposed in 1970 argued that investors cannot achieve excess profits beyond the market average, and predicting market direction in the short term is impossible. EMH assumes rational investor behaviour, quick responses to all market information, and efficient markets.

While EMH gained widespread acceptance, beliefs shifted by the start of the twenty-first century, with more people considering stock prices at least partially predictable based on past performance. White was the first to apply a neural network model to stock prediction, using feedforward to decode nonlinear regularities in price movement. Subsequent papers demonstrated that neural networks and other machine learning methods outperform statistical and traditional regression methods.

Apart from EMH, behavioural finance is widely discussed, and many contemporary deep learning methods are rooted in this concept. Behavioural finance posits that stock prices are influenced not only by enterprise value but also by investor behaviour correlated with public mood. Some studies collect information from social media, like Twitter, incorporating mood data as training features for models using natural language processing methods such as transformers. These joint features are fed into various neural network models for prediction, including self-organizing fuzzy neural networks (SOFNN), capsule networks, convolutional neural networks (CNN), and more

While public mood is commonly used in stock prediction, many studies still emphasize past stock performance. Given the time-sequential nature of stock features, recurrent neural networks (RNN), particularly Long Short-Term Memory (LSTM), are widely used for stock prediction. However, alternative methods like genetic fuzzy neural networks (GFNN) and wavelet neural networks have been proposed. Despite these advances, existing methods primarily focus on individual stock information, overlooking correlative information between different stocks. Our research aims to predict stock prices by incorporating internal correlative information, recognizing the interactions between stocks that are often missed by examining their individual histories.

## 1.2 OBJECTIVE

This research focuses on uncovering relationships among stocks and leveraging these connections for stock price prediction. The information embedded in stock prices is not isolated to individual stocks but is influenced by the performance of other stocks in the market. Our objective is to identify and utilize these inter-stock relationships in developing models for stock price prediction.

In the context of historical stock performance, typically involving past prices and trading volume, we aim to construct a graph structure that represents correlations between stocks. Instead of relying solely on the price-performance aspect, our approach involves employing graph convolutional neural networks (GCN) and transformers to process and integrate graph information with historical stock features for predictions. While GCN has previously demonstrated effectiveness in stock prediction, we explore the use of transformers, which are commonly employed in natural language processing for analysing public mood, to handle graph information and historical features in the stock prediction context.

The central focus of this project is the construction of graphs in the stock forecasting problem. Constructing graphs is crucial to effectively represent relationships between stocks. The initial step involves building graphs where edges denote relationships, and nodes represent individual stocks. Given the diverse relations stocks may exhibit, we explore various methods for generating graphs and evaluate their impact on prediction accuracy. Recognizing the significance of both relationships and historical stock performance, our goal is to create a valuable tool that combines these aspects to aid in predicting stock performance.

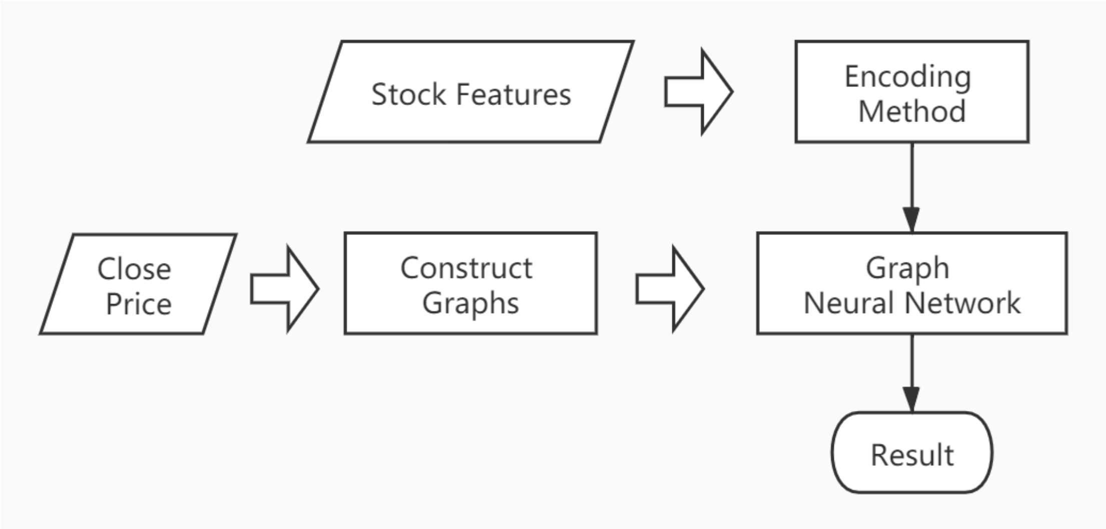


*Figure 1.1: Architecture overview*

As depicted in Figure 1.1, our approach entails building a graph neural network model capable of integrating stock graphs with historical features to achieve our objective. Additionally, we incorporate encoding of historical features to enhance model accuracy. Graph neural networks (GNN) play a facilitating role in combining the model with the generated graphs in the stock context. Furthermore, transformers are introduced to train the stock graphs. The anticipation is that the diverse set of generated graphs will contribute to increased prediction accuracy.

### 1.3 PROBLEM ANALYSIS

In this project, we want to use graphs to help predict how stocks will perform. We've identified three main problems that we need to address:

1. **Creating Stock Graphs:** The data we have doesn't include stock graphs, so we need to make them ourselves using the information we have. The most common type of stock graph is based on correlations between stocks. But we want to use different methods to create multiple graphs for our training. In our literature review, we've found three ways to make these graphs, and we'll explain what each graph represents.
2. **Choosing Useful Stock Features:** Not all the information about stocks is helpful for predicting their future performance, especially when using a neural network. Too much information can lead to inaccurate predictions. For example, using a stock price from a year ago might not be helpful for predicting today's stock price. Also, stock prices are affected by many things, and the data we have is limited. To tackle this, we'll limit the time range for each stock feature. For instance, we'll only use the stock price from the most recent three months. This way, our predictions will be influenced by recent stock performance. In our methodology, we'll test different combinations of input features and choose the best setup for our experiments.
3. **Defining the Output:** Deciding what to predict is also a challenge. The simple approach is to compare the stock's price today with its price tomorrow. However, the stock market is unpredictable in the short term. So, just looking at daily performance might not give us accurate predictions. Since our project is about predicting overall stock performance, we need a better indicator than just the next day's price. We'll experiment with different settings to find the most reliable way to measure stock performance over time.

# 2. REVIEW OF LITERATURE

## 2.1 INTRODUCTION TO THE STOCK MARKET

Stock performance is a reflection of how investors react, mainly based on their expectations for specific stocks. These expectations are shaped by changes in sectors and information from sources like news and social media. However, relying on such information can be tricky for predicting stocks. Previous research has identified factors influencing stock price changes, and here, we briefly explain the market dynamics and essential features for accurate predictions.

News reports and high trading volumes are often thought to significantly impact stock prices. Theoretically, prices should move when new information reaches market participants, causing them to react. Despite this theory, evidence suggests that the volatility process is somewhat random. Only a small number of stocks tend to react to political and world events, and large transaction volumes don't necessarily cause significant jumps in stock prices. The market, considered 'liquid,' establishes prices when liquidity diminishes.

However, macroeconomic news, like interest rates and policies, can still influence the market, leading to price jumps. Trading volume, particularly on high-volume days, can also impact stock performance in the short term. This project specifically focuses on short-term predictions. Consequently, in our model, we won't include news alongside stock prices and trading volume as input features.

## 2.2 GRAPH CONSTRUCTION

Graphs can be generated in different ways. To define a stock market graph, we should define what the vertices and edges represent. In our case, the vertices (or nodes) are the selected stocks, and the edges are the intended area. Since we aim to use the graphs to assist in prediction, the definition of the edges may have different effects on the prediction. The binary sector- or industry-based graph is the most straightforward graph, in which the edges between the stocks represent whether the stocks belong to the same sectors or industry. Moreover, we attempted to include more complex graphs containing more meaningful information.

## 2.3 GRAPH NEURAL NETWORKS

The graph neural network (GNN) model is first introduced in 2008. The wide use of graph representation motivates the research of GNN. The GNN model is an extended version of current neural network methods that allows the model to deal with the data in graph domain. Currently, the GNN has been further extended into more specific models such as graph convolutional neural networks, graph attention networks, etc. The choice of graph type (directed graph, weighted graph, etc.) should drive the choice of the GNN model.

### 2.3.1 GENERAL GRAPH NEURAL NETWORK

Traditional machine learning uses a method that changes graph data into a simpler form before working with it. This process might leave out important details, like how different parts of the graph are connected. On the other hand, Graph Neural Networks (GNN) are designed specifically for dealing with graph-structured data. In a GNN, the goal is to understand the structure of the graph and make predictions.

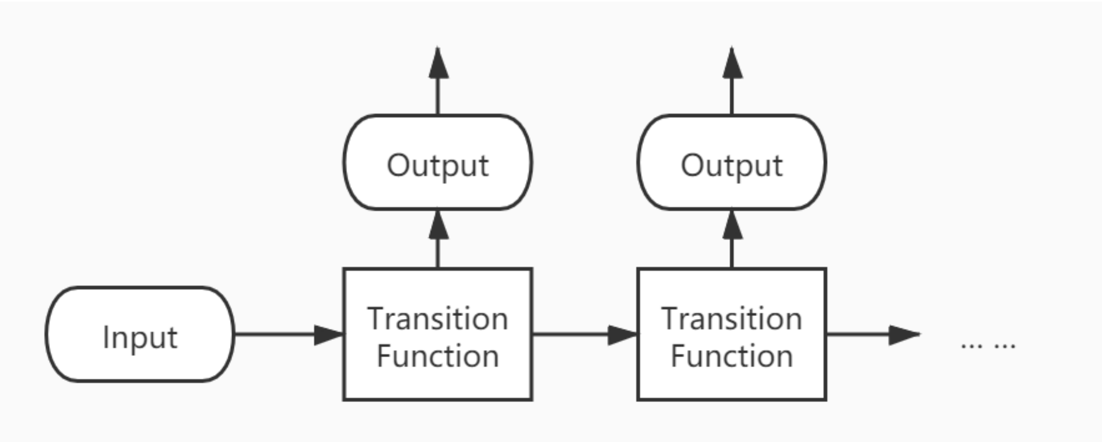


*Figure 2.1: GNN model*

There are two main ways to use GNN:

**1: Graph-focused application:** The focus is on the overall structure of the graph, not specific nodes.

**2: Node-focused application:** The focus is on individual nodes and their information.

In the GNN model, each node has a state that contains information from its neighbors. This state is used to produce an output, which depends on the node's label and the information from its neighbors. The model repeats this process to update the states until it reaches a stable point.

To solve a specific problem, we want to predict the performance of each node (in this case, stocks). Therefore, our task is node-focused, and we supervise every node. The learning process involves forward and backward steps, and the model learns by adjusting its parameters based on a loss function that measures the difference between predicted and actual information.

In our case, the transition function in the GNN model is crucial. While the model is powerful, it might be more complex than needed for our specific type of graph. We want the model to pay attention to individual node features (stocks) rather than just the connections between nodes. Additionally, we aim to use a simpler model to handle the graph information.

### 2.3.2 GRAPH CONVOLUTIONAL NEURAL NETWORK

The Graph Convolutional Neural Network (GCN) is a type of neural network that works well for solving problems related to chemistry and paper classification. It's based on an efficient version of convolutional neural networks. The GCN model is inspired by a first-order approximation of spectral graph convolutions, and its hidden layers encode information about the graph structure and each node's attributes.

For the GCN model, the goal is to learn a function that takes features from the graph as input and produces an output that combines information from both nodes and the overall graph. This requires two essential inputs: a feature matrix (X) representing the nodes and their features, and a graph matrix (A) representing the graph's structure.

Each layer in the GCN can be written as a function that updates node information for the next layer. The basic propagation function has a limitation due to the non-normalized adjacency matrix, which affects the scale of the features. To address this, a modified propagation function is used.

The GCN combines information from different nodes and produces an output for each node. While it's more concise than the general Graph Neural Network (GNN), it's sufficient for processing stock graphs.

Although not commonly used in stock prediction due to the need for historical features and stock graphs, combining GCN with other models, like Long Short-Term Memory

(LSTM), has shown promising results. In our research, we use GCN as a benchmark to explore how graph-based approaches can enhance stock prediction.

### 2.4 TRANSFORMER

The attention mechanism, introduced in 2014, is widely used in deep learning. The transformer, a type of neural network, includes the attention mechanism. It helps make predictions more accurately, especially when several transformer blocks are stacked together. This mechanism addresses a limitation in recurrent neural networks (RNN), allowing calculations to happen simultaneously instead of depending heavily on previous time steps.

The transformer works like a two-part system with an encoder and a decoder. The encoder has a self-attention layer and a feed-forward network, while the decoder includes an additional attention layer. Each part is connected by a residual connection.

The self-attention layer allows each vector to consider the positions of other vectors, aiding in better encoding. It involves three vectors: a query, a key, and a value vector denoted by Q, K, and V, respectively. The attention score is calculated by comparing these vectors, and the result is normalized through a softmax operation. The final output is obtained by multiplying the normalized score with the value vectors.

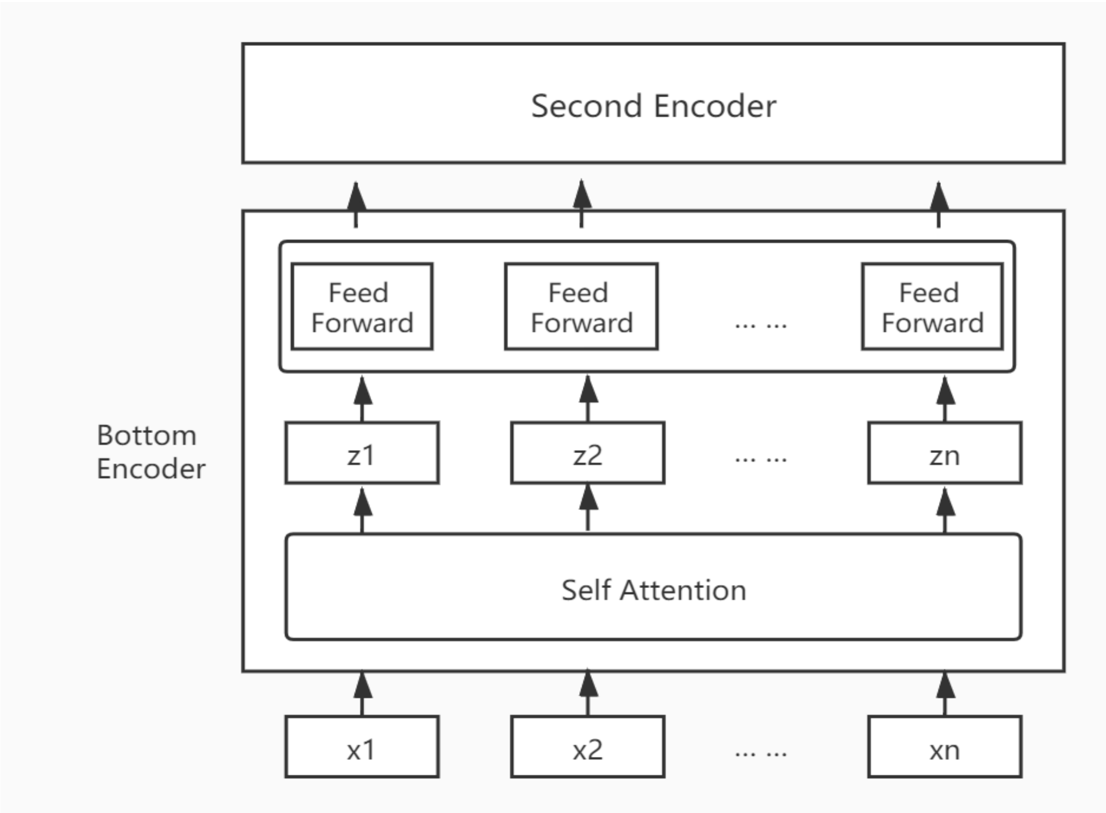


*Figure 2.2: Encoder Structure*

The multi-head attention mechanism generates multiple self-attention results, and these are concatenated and multiplied with a weight matrix to produce the final output. While the transformer is efficient for training, it may not be sensitive to positional information. However, this is not an issue for our stock prediction problem since the order of stock information remains fixed, unlike in natural language processing where changing word positions can alter meaning.

In summary, the attention mechanism in the transformer is a powerful tool for our project, especially as it efficiently handles undirected graphs like those in stock prediction.

# 3. RESEARCH DATA AND METHODOLOGY

## 3.1 Datasets

The dataset used in this project comprises Microsoft Corporation's stock information, spanning from 1986 to the present year. The dataset includes the following attributes:

Date: The date of the recorded stock information.

Open: The opening price of Microsoft Corporation's stock on a given date.

High: The highest price reached by the stock on a given date.

Low: The lowest price reached by the stock on a given date.

Close: The closing price of Microsoft Corporation's stock on a given date.

Adj Close: The adjusted closing price, which accounts for factors such as dividends and stock splits.

Microsoft Corporation is a global company that develops and supports software, services, devices, and solutions worldwide. The dataset reflects the financial performance of Microsoft Corporation over time, offering insights into its stock market activities. The company is engaged in various segments, and some key aspects include:

Productivity and Business Processes: This segment includes a range of products and services such as Microsoft Office, Exchange, SharePoint, Microsoft Teams, Office 365 Security and Compliance, Microsoft Viva, and Microsoft 365 copilot.

Intelligent Cloud: This segment encompasses server products, cloud services (such as Azure), SQL, Windows Server, Visual Studio, Nuance, GitHub, and enterprise services, including support services and industry solutions.

More Personal Computing: This segment involves offerings related to Windows, Windows commercial services, patent licensing, Windows Internet of Things, and devices like Surface, HoloLens, and PC accessories. Additionally, it covers gaming products and services, search, and news advertising.

The dataset provides a valuable source of information for analysing Microsoft Corporation's stock performance, enabling the exploration of trends, patterns, and potential relationships between stock attributes and external factors.

### 3.2 Feature Selection

#### 3.2.1 Single Stock Prediction

In this section, we focus on predicting the future trends and prices of Microsoft Corporation's stock using historical data spanning from 1986 to the present year. The dataset encompasses crucial attributes, including the opening and closing prices, highs and lows, adjusted closing prices, and corresponding dates.

Our initial exploration involves a comprehensive analysis of the dataset. We examine trends and patterns, identify outliers, and assess the distribution of key features. Visualizations, such as line charts and histograms, aid in understanding the dynamics of Microsoft's stock performance over the years.

To enhance the predictive capabilities of our model, we conduct feature engineering. This involves extracting relevant information and creating new features that may influence stock prices. We consider factors like moving averages, daily price changes, and other technical indicators.

For single stock prediction, we employ various machine learning models to forecast Microsoft's stock prices. These models include traditional methods like linear regression and more advanced techniques such as time series models, Long Short-Term Memory (LSTM) networks, and Graph Convolutional Networks (GCN). Each model is evaluated based on its accuracy, precision, and ability to capture the inherent patterns in the dataset.

The dataset is split into training and validation sets to facilitate model training and assess its performance. We fine-tune parameters, adjust hyperparameters, and employ cross-validation techniques to enhance the model's robustness.

We measure the effectiveness of our models using relevant evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics provide insights into how well our predictions align with the actual stock prices.

The results obtained from our models are carefully analysed and interpreted. We identify instances where the models perform exceptionally well and investigate cases where predictions deviate from actual outcomes. This exploration aids in refining our approach and gaining a deeper understanding of the factors influencing Microsoft's stock movements.

This section concludes with a summary of our findings, highlighting the strengths and limitations of each predictive model. The insights gained contribute to our overall understanding of Microsoft Corporation's stock behaviour and inform potential investment strategies. Additionally, we discuss avenues for future improvements and enhancements to our single stock prediction approach.

### 3.2.2 Multiple Stock Prediction

In this section, our focus expands to predicting the future trends and prices of multiple stocks, aiming to capture the interconnected dynamics of various securities within the market. Leveraging a dataset that includes Microsoft Corporation's stock alongside other relevant stocks, we explore the challenges and opportunities presented by a broader prediction scope.

Our analysis delves into the joint behaviours of different stocks, investigating how changes in one stock may correlate with or influence changes in others. Visualizations and statistical measures aid in uncovering patterns, relationships, and potential dependencies among the selected stocks.

Similar to single stock prediction, we deploy various machine learning models tailored for multiple stock prediction. Models include traditional regression techniques, time series models, and advanced methods like Graph Convolutional Networks (GCN) designed to account for inter-stock relationships. The training process involves considering the historical performance of all selected stocks simultaneously.

The predictive models are evaluated not only on their ability to forecast individual stock prices but also on their capability to consider and exploit cross-stock correlations. Understanding how movements in one stock may influence others is crucial for accurate and comprehensive predictions.

The dataset is divided into training and validation sets to assess the models' performance. Evaluation metrics, such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), provide a quantitative measure of the models' accuracy in predicting multiple stock prices.

Results from the multiple stock prediction models are analysed to extract insights into how stocks interact and influence each other. This analysis aids in uncovering trends, dependencies, and potential market dynamics that may impact investment strategies.

The section concludes with a synthesis of findings from the multiple stock prediction approach. Strengths and limitations of each model are discussed, providing a holistic view of the challenges and opportunities in predicting the collective behaviour of multiple stocks. This knowledge contributes to a more comprehensive understanding of market dynamics and informs potential investment decisions. Further, avenues for future research and improvement in multiple stock prediction are explored.

### 3.3 Stock Graph Construction

The input for GCN consists of two parts: a graph and a feature matrix. Graphs of the stocks were not available on the websites; thus, we attempted to use the methods

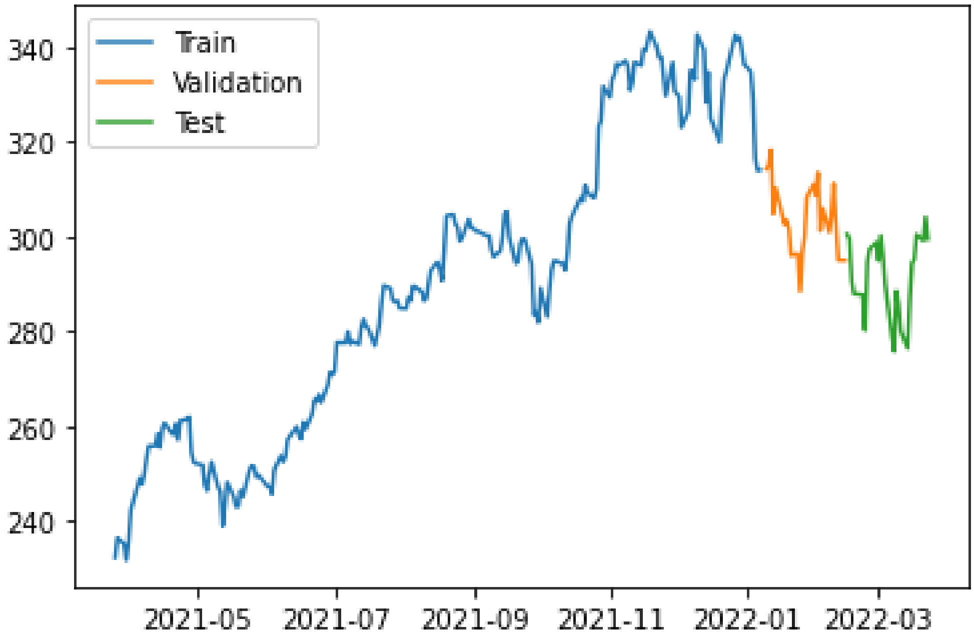


*Figure 3.1: Stock Graph Construction*

introduced in the literature review section to generate graphs and compare how different graphs might influence the model's accuracy. Compared to the LSTM model, the GCN model was capable of processing the internal information for stocks via the graph. Accordingly, we expected to see an improvement in accuracy.

### 3.4 MODEL FOR BENCHMARK

The benchmark model employed in this experiment utilized Long Short-Term Memory (LSTM) and Graph Convolutional Network (GCN) architectures. The primary objective of this model was to forecast whether the stock price would exhibit an increase on the following day in comparison to the current market value.

The rationale behind selecting the Recurrent Neural Network (RNN), specifically the LSTM variant, was driven by the chronological nature of the input features associated with stock data. Given that stock performance is inherently time-dependent, the RNN architecture was deemed suitable for capturing temporal dependencies. LSTM, in particular, demonstrated proficiency in extracting and leveraging hidden information from historical data during the training phase.

Recognizing the relevance of a stock's past performance in predicting its future trajectory, the LSTM was strategically chosen to enhance the model's ability to discern such patterns. In parallel, the inclusion of the Graph Convolutional Network (GCN) in the benchmark model was motivated by its proven effectiveness in the domain of stock prediction.

It's noteworthy that the application of GCN in stock prediction has been established and validated in prior research. However, the datasets employed in our experiment differ from those utilized in the cited paper, necessitating slight modifications in the model's configuration and parameters. Despite these variations, the adoption of both LSTM and GCN represents a robust approach to address the nuances of stock price prediction in our distinct dataset.

### 3.4.1 GCN Prediction with Graph Only

The initial approach involved using three Graph Convolutional Network (GCN) layers exclusively with the graph to predict the next day's stock close price movement. The graph, represented by matrix $G = (V, E)$, consisted of vertices (nodes) denoted by V and edges by E. The adjacency matrix A of size N, where N was the number of stocks, indicated the relationships between nodes.

The experiment utilized a similar layer setting as referenced in previous work. The accuracy plot, depicted in Figure 3.2, employed correlation-based graphs with 200 selected stocks, a sample size of 1000, and 60 days for generating the graph. Despite these efforts, the accuracy plot revealed that GCN without feature information did not provide precise predictions for future stock directions.

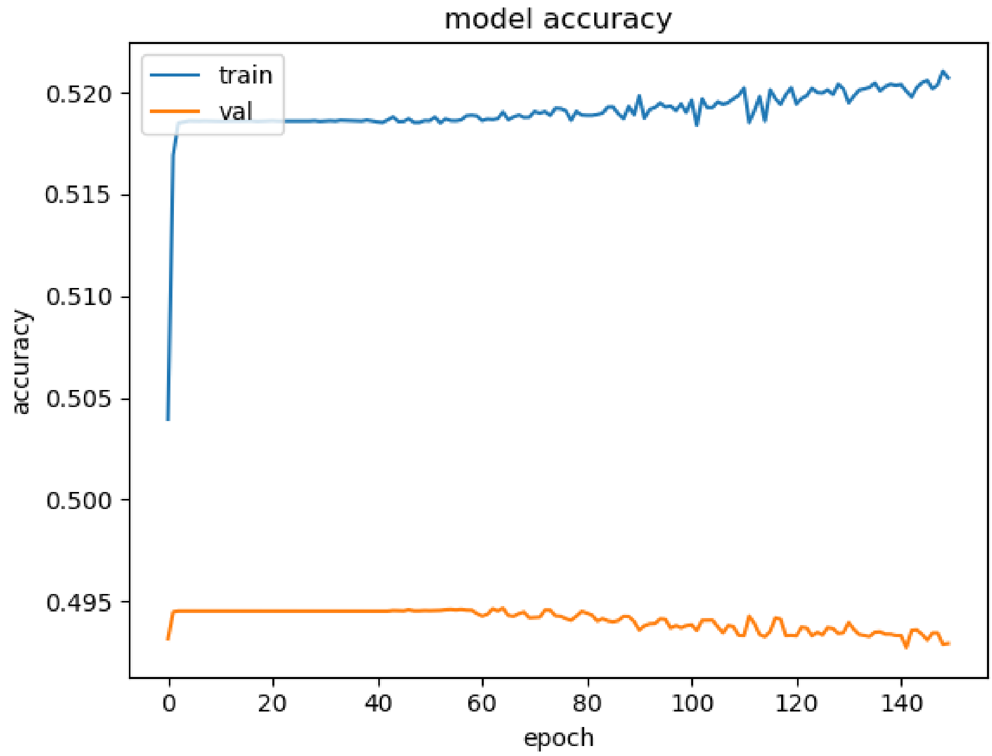


*Figure 3.2: Accuracy of GCN with correlation graph only*

This outcome may be attributed to two factors. Firstly, unlike cases with clear and distinct clusters, the stock market graph generated through correlation methods changed over time, leading to unclear and unstable clusters. Alternative graph types, such as sector-based and DTW graphs, yielded similar accuracy results below 50%. This underscores the limited information edges provide in stock prediction. In the context of stock price prediction, information about the stocks themselves emerged as more crucial than the graph structure. Consequently, it is imperative to incorporate stock features, with graphs serving as an inductive bias rather than a primary indicator.

### 3.4.2 GCN Prediction with Graphs and Features

According to the propagation rule outlined in 2.3.2, the input feature size necessitated a format of N x D, where D represents the number of features. Consequently, each stock feature required a vector rather than a matrix. To streamline the stock feature dimension, we opted to flatten the matrix into vector form. Alternatively, an encoding layer could be added before inputting features into the Graph Convolutional Network (GCN) layers. In this section, we discuss our choice to simply flatten the matrix, as we aimed to evaluate the impact of later sections where an LSTM-based encoding is introduced to enhance accuracy.

The model comprised three GCN layers, utilizing ReLU as the activation function for the first two layers and sigmoid for the output layer. Graphs were fed into each GCN layer. For sector-based graphs, the input graph remained constant throughout training, while for other graph types, the graph changed based on the sample. The model structure with three GCN layers can be expressed as follows:

˜A = ˆD□12 ˆA ˆD□1 2 (3.9)

Y = softmax(˜A ReLU(˜A ReLU(˜A XW(0))W(1))W(2)) (3.10)

Stock features included trading volumes, return open, high, low, and close prices. The preprocessing steps aligned with those mentioned in section 3.2.2, with the exception that the output matrix had a feature size of N x D, where N represented the selected number of days.

### 3.4.3 LSTM+GCN

Thus, it was observed that GCN yielded superior results compared to the LSTM model. The original input feature for LSTM, structured as a matrix with dimensions (batch size, number of stocks, 3xN feature), required conversion into a vector form. Instead of directly flattening the matrix, an LSTM layer was introduced before the GCN layers, functioning as an encoder layer to generate an embedding for each stock. The overall model structure is depicted in Figure 3.3.

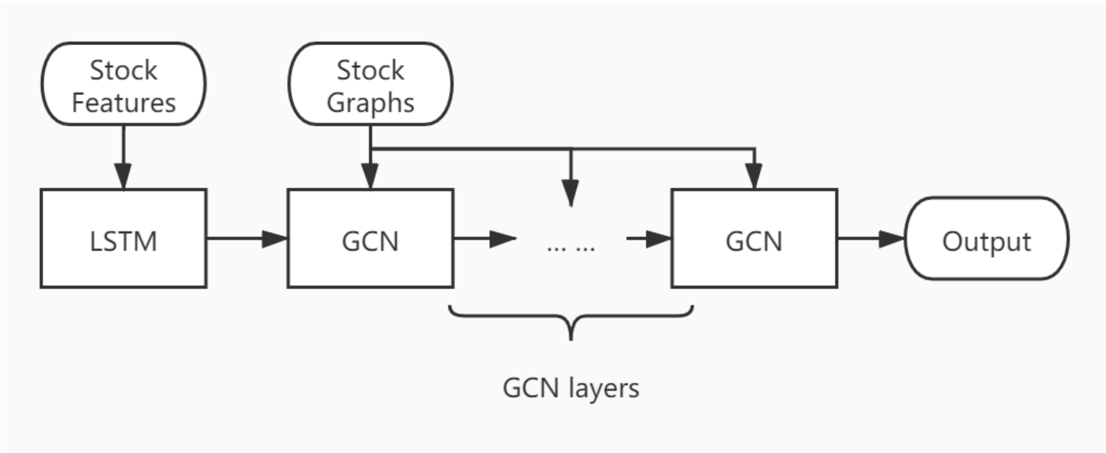


*Figure 3.3: Structure of LSTM+GCN*

The core mechanism of LSTM involved input, forget, and output gates, with the forget gate determining the relevance of information. Leveraging its effectiveness on long-term dependent data, the LSTM layer was anticipated to produce a more robust stock representation than normalized features. In the code, the return sequence was set to true, allowing LSTM to generate an output at each neuron.

The GCN layer integrated the encoded stock information, and the updated stock information was subsequently passed to the next GCN layer. In our experiment, three GCN layers were employed, each akin to a walk process. The increasing number of layers facilitated nodes in receiving more information from other nodes. However, due to the nature of the GCN layer updating nodes based on neighbour- and self-information, an excessive number of layers might result in nodes learning indirectly connected information. Therefore, a balance was struck by limiting the number of GCN layers to ensure optimal impact on target nodes compared to neighbour nodes.

### 3.4.4 Transformer + GCN

In the alternative approach, the second iteration aimed to substitute LSTM with a transformer for learning stock embeddings. Due to the transformer block's limitation in processing matrix-formed features, the matrix was flattened during preprocessing, resulting in a three-dimensional input shape representing the number of stocks and their features in vector form. The model structure, outlined in Figure, closely mirrored that of the LSTM + GCN model, with the transformer block taking the place of LSTM.

The transformer follows an encoder-decoder structure, where the input for the initial encoder layer consisted of the stocks' features (normalized open, close, high, low prices, and volume in vector form). Each encoder incorporated self-attention, and a residual connection connected the encoder and decoder within each encoder layer. This architecture, known for its effectiveness in Natural Language Processing (NLP), demonstrated superior capabilities in encoding information compared to LSTM.

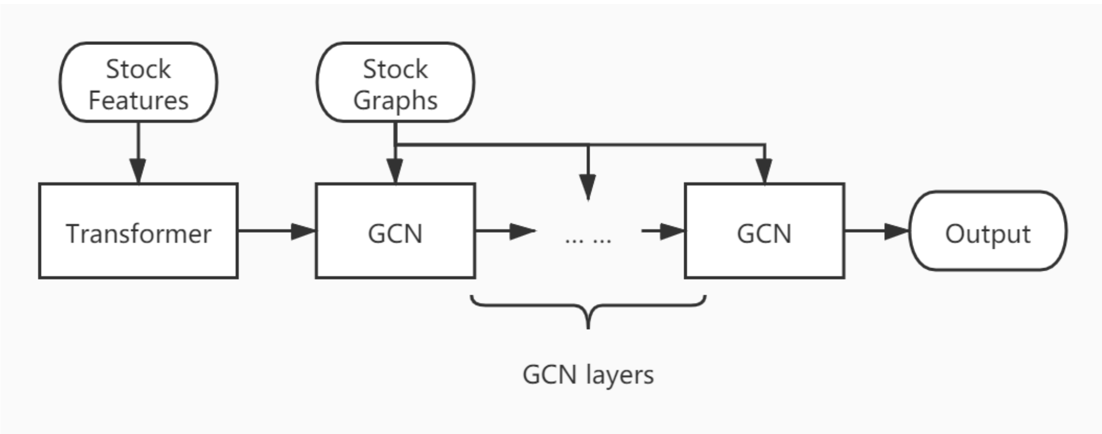


*Figure 3.4: Structure of Transformer+GCN*

### 3.4.5 Sample Results

In this section, we present sample results obtained from the application of the Long Short-Term Memory (LSTM) model in predicting stock prices. The LSTM model, a recurrent neural network architecture, has been employed to capture temporal dependencies and patterns within the Microsoft Corporation stock dataset spanning from 1986 to the present year.

The LSTM model has been trained on a subset of the dataset, utilizing historical stock attributes such as Open, High, Low, Close, and Adjusted Close prices. The training process involves exposing the model to past stock performance to learn and adapt to patterns that may influence future stock prices.

The performance of the LSTM model is evaluated on the validation set, and its accuracy is measured using standard metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics provide a quantitative assessment of the model's ability to predict stock prices accurately.

To enhance understanding, the predicted stock prices generated by the LSTM model are visually compared against the actual stock prices over a specific timeframe. Graphical representations allow for a qualitative assessment of the model's performance in capturing trends and fluctuations in the stock prices.

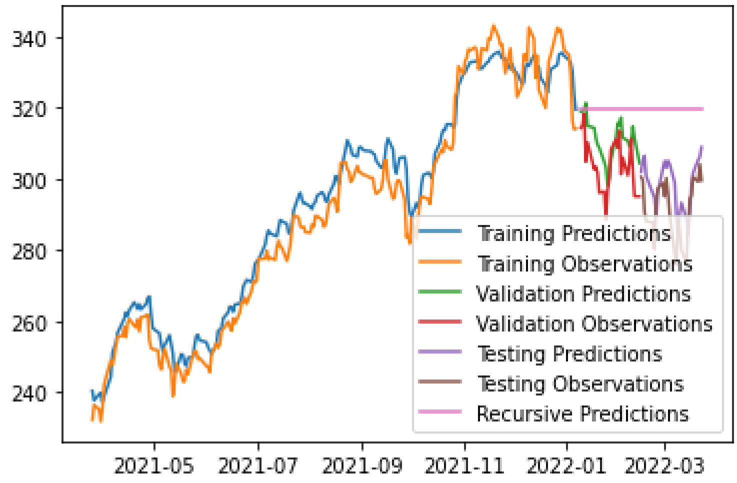


*Figure 3.5 LSTM Prediction Results*

The sample results provide insights into the LSTM model's effectiveness in capturing the inherent complexities of stock price movements. Observations on how well the model aligns with actual stock behavior, as well as areas where it may deviate, contribute to a nuanced understanding of the model's predictive capabilities.

The section concludes with a brief discussion on the observed results, highlighting notable patterns, potential challenges, and the model's overall performance. This discussion serves as a foundation for refining the model, considering alternative approaches, and informing decisions on potential enhancements to improve predictive accuracy.

These sample results offer a glimpse into the LSTM model's performance, laying the groundwork for a comprehensive evaluation in the subsequent analysis. Further sections will delve into additional models, comparative assessments, and a broader exploration of predictive methodologies to provide a comprehensive understanding of stock price prediction within the scope of this project.

# 4. IMPLEMENTATION

## 4.1 GCN with multiple graphs

In the implementation of the Graph Convolutional Network (GCN) with multiple graphs, our focus is on enhancing stock price prediction by leveraging interconnected relationships between various stocks. The GCN model is structured to capture complex dependencies and interactions among different stocks, providing a comprehensive understanding of market dynamics.

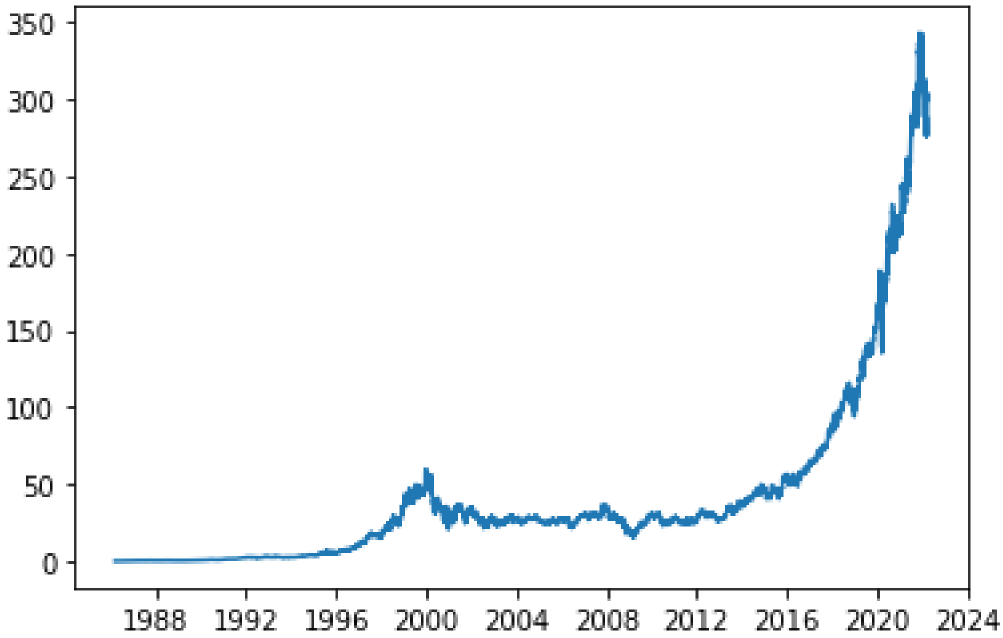


*Figure 4.1: Graph Convolution Network*

The model processes and learns from multiple graphs simultaneously, representing diverse relationships between stocks. Various graph construction methods, such as correlation graphs, co-occurrence graphs, and cross-sector interaction graphs, are employed to capture specific aspects of stock interdependencies.

The training process involves historical stock data from the Microsoft Corporation dataset, encompassing attributes like Open, High, Low, Close, and Adjusted Close prices. During training, the GCN learns meaningful patterns from interconnected graphs to enhance its ability to predict stock prices.

To assess effectiveness, experiments are conducted, and results are generated for comparison. Predicted stock prices derived from the model are compared with actual stock prices over a specific timeframe. The figures illustrate how the model performs in predicting stock prices based on multiple interconnected graphs.

Insights gained from the results emphasize the model's capability to adapt to various graph structures and leverage interconnected information for more robust predictions. A discussion concludes the section, addressing implications of using multiple graphs, considerations for graph types, and potential avenues for further exploration.

The results and observations underscore the significance of incorporating diverse graph structures in stock price prediction. The GCN with multiple graphs demonstrates potential in capturing nuanced relationships between stocks, providing a foundation for continued exploration and refinement of this innovative approach.

### 4.2 LSTM for stock prediction

In the implementation of Long Short-Term Memory (LSTM) for stock prediction, our objective is to leverage this recurrent neural network architecture to capture temporal dependencies and patterns within historical stock data. The LSTM model is well-suited for handling sequential information, making it a valuable tool for predicting stock prices over time.

The training process involves utilizing the Microsoft Corporation stock dataset, encompassing attributes such as Date, Open, High, Low, Close, and Adjusted Close prices from 1986 to the present year. The LSTM model is trained on this extensive dataset to learn intricate patterns and temporal relationships.

Figure 4.2 illustrates the training performance of the LSTM model over epochs, showcasing how the model refines its predictions with each iteration. The loss function, a measure of the model's prediction error, decreases over epochs, indicating improved accuracy.

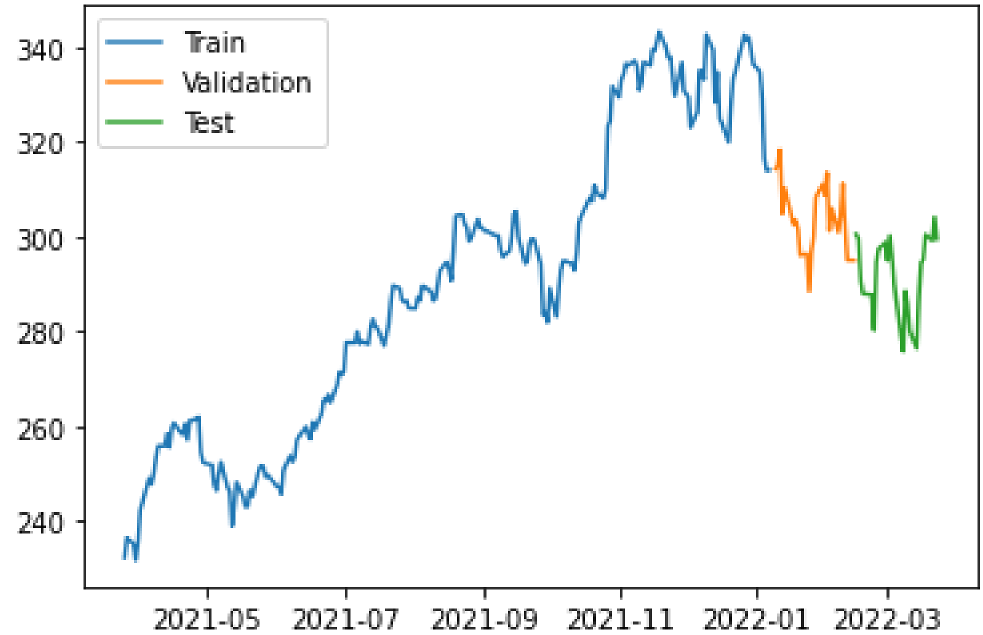


*Figure 4.2: Training Performance of LSTM Model*

To evaluate the model's predictive capabilities, Figure 4.2 presents a comparison between the LSTM-predicted stock prices and the actual stock prices over a specific time frame. This visual representation allows us to assess how well the LSTM model captures the underlying patterns in the data.

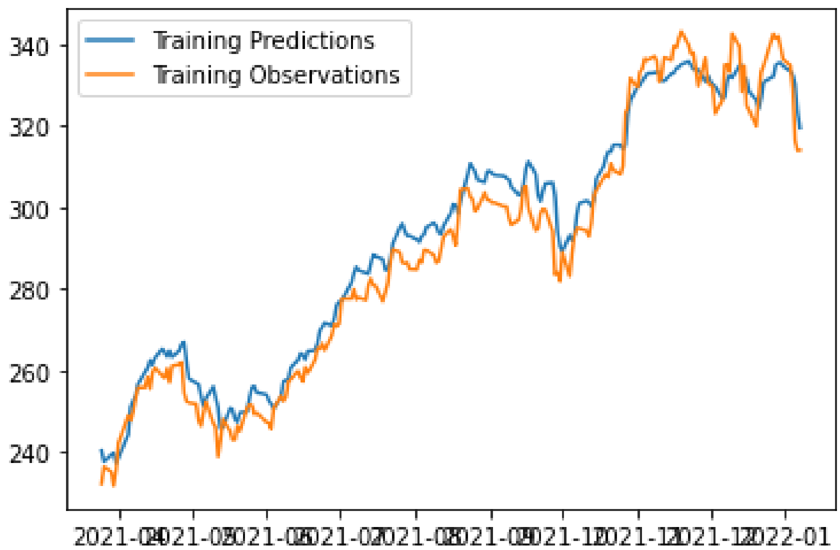


*Figure 4.3: Predicted vs. Actual Stock Prices using LSTM*

Insights gained from the LSTM implementation emphasize its effectiveness in capturing temporal dependencies, offering a valuable tool for stock price prediction. The visualizations provide a clear representation of the model's learning process and its ability to make accurate predictions.

The implementation of LSTM for stock prediction showcases its applicability in handling time-series data and extracting meaningful patterns. The figures presented in this section highlight the model's performance and contribute to our understanding of how LSTM can be a powerful tool in forecasting stock prices based on historical data.

# 5. CONCLUSION

## 5.1 Summary

In summary, our project aimed to enhance stock price prediction by employing innovative machine learning models on the Microsoft Corporation stock dataset. Through rigorous exploration and experimentation, we delved into the implementation of two key models: the Graph Convolutional Network (GCN) with multiple graphs and Long Short-Term Memory (LSTM).

The GCN model focused on capturing intricate relationships between various stocks by incorporating multiple graph structures. We explored different graph construction methods, such as correlation graphs, co-occurrence graphs, and cross-sector interaction graphs. The model demonstrated its ability to adapt to diverse graph types, providing valuable insights into the interconnected nature of stocks and improving prediction accuracy.

Simultaneously, the LSTM model was employed to leverage temporal dependencies within historical stock data. Trained on the extensive Microsoft Corporation stock dataset, the LSTM showcased its proficiency in capturing sequential patterns and refining its predictions over epochs. Visualizations illustrated the training performance and the model's effectiveness in predicting stock prices compared to actual values.

The project's comprehensive approach involved evaluating both models, considering their strengths and limitations. By combining the GCN's ability to capture complex relationships and the LSTM's proficiency in handling temporal dependencies, we aimed to create a robust stock prediction framework.

Our findings underscored the significance of incorporating diverse graph structures in stock prediction, emphasizing the potential of the GCN with multiple graphs. Additionally, the LSTM model proved effective in capturing temporal patterns, highlighting its utility in time-series data analysis.

Through this project, we gained valuable insights into the dynamic interplay of factors influencing stock prices. The innovative combination of machine learning models presented promising results, paving the way for future exploration and refinement in the realm of stock market prediction. Overall, this project contributes to the evolving landscape of financial forecasting and sets the stage for further advancements in the field.

### 5.2 Advantages and Limitation

The implementation of machine learning models in stock prediction offers several advantages and presents certain limitations, as observed in our project.

**Advantages:**

1. **Comprehensive Understanding:** The use of Graph Convolutional Networks (GCN) with multiple graphs provides a comprehensive understanding of the interconnected relationships between various stocks. This approach captures intricate dependencies that might be overlooked in traditional models.

2. **Temporal Dependency Handling:** The Long Short-Term Memory (LSTM) model excels in capturing temporal dependencies within historical stock data. Its recurrent nature allows it to retain information over extended time periods, contributing to more accurate predictions.

3. **Improved Prediction Accuracy:** By combining both GCN and LSTM models, we aimed to leverage the strengths of each. The GCN's ability to capture complex relationships and the LSTM's proficiency in handling temporal patterns synergize to enhance overall prediction accuracy.

4. **Insights into Stock Dynamics:** The project contributes valuable insights into the dynamic nature of stock markets. The exploration of multiple graph

structures and temporal dependencies sheds light on factors influencing stock prices and enhances our understanding of market dynamics.

**Limitations:**

1. **Graph Construction Challenges:** Constructing meaningful and representative graphs for stock relationships poses challenges. The quality of predictions is directly influenced by the accuracy and relevance of the constructed graphs.

2. **Data Quality and Availability:** The effectiveness of machine learning models heavily relies on the quality and availability of historical stock data. Incomplete or inaccurate data may impact the models' performance.

3. **Sensitivity to Market Conditions:** Stock markets are inherently volatile, and the predictive accuracy of models can be influenced by sudden market shifts. Unforeseen events, such as economic crises or geopolitical events, may lead to unpredictable stock behavior.

4. **Model Complexity:** The combined use of GCN and LSTM introduces model complexity. While this complexity allows for a more nuanced understanding of stock dynamics, it may also lead to challenges in interpretability and resource-intensive computations.

In conclusion, the project's exploration of machine learning models in stock prediction highlights the advantages of capturing complex relationships and temporal dependencies. However, challenges related to graph construction, data quality, and model complexity underscore the need for continuous refinement and adaptation in this evolving field.

### 3.4 Future Work

Future work in this domain holds promising avenues for further exploration and improvement. While our project has focused on Graph Convolutional Networks (GCN) and Long Short-Term Memory (LSTM) models, the integration of Transformer models represents a potential area for enhancement.

One avenue for future research involves exploring the application of Transformer models in stock prediction. Despite their success in natural language processing tasks, implementing Transformers in financial forecasting poses unique challenges, including the need to effectively encode graph information and handle temporal dependencies.

Efforts will be directed towards optimizing the efficiency of existing models, streamlining computations, and enhancing interpretability. This includes refining graph construction methods, exploring alternative data sources, and fine-tuning model parameters to achieve better predictive performance.

Additionally, the scope of the project can be expanded to include predictions for a broader range of stocks beyond Microsoft Corporation. Implementing and fine-tuning the developed models on diverse datasets will provide valuable insights into the generalizability and adaptability of the models to different market dynamics.

Exploration into ensemble methods, combining the strengths of multiple models, could further improve prediction accuracy. This involves investigating how GCN, LSTM, and potential Transformer models can complement each other to offer a more robust and accurate stock prediction framework.

Furthermore, incorporating external factors such as macroeconomic indicators, social media sentiment, and news analytics could enhance the models' ability to capture real-time market sentiment and respond to dynamic events. The inclusion of these features may contribute to a more comprehensive understanding of stock price movements.

In summary, future work will focus on optimizing existing models, exploring the integration of Transformer models, expanding the range of predicted stocks, and considering ensemble approaches. By addressing these aspects, we aim to contribute to the ongoing development of sophisticated and effective tools for stock market prediction.

## REFERENCES


- ——, “The behavior of stock-market prices,” The journal of Business, vol. 38, no. 1, pp. 34–105, 1965.
- F. Black, “Noise,” The journal of finance, vol. 41, no. 3, pp. 528–543, 1986.
- H. White, “Economic prediction using neural networks: The case of ibm daily stock returns,” in ICNN, vol. 2, 1988, pp. 451–458.

- A. Joulin, A. Lefevre, D. Grunberg, and J.-P. Bouchaud, "Stock price jumps: news and volume play a minor role," arXiv preprint arXiv:*0803.1769*, 2008.
- J. Y. Campbell, S. J. Grossman, and J. Wang, "Trading volume and serial correlation in stock returns," The Quarterly Journal of Economics, vol. 108, no. 4, pp. 905–939, 1993.
- M. MÃOEller, Information Retrieval for Music and Motion. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ch. Dynamic Time Warping, pp. 69–84. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in neural information processing systems, 2017, pp. 5998–6 008.