Spring 5-20-2019

STOCK MARKET PREDICTION USING ENSEMBLE OF GRAPH THEORY, MACHINE LEARNING AND DEEP LEARNING MODELS

Pratik Patil
San Jose State University

STOCK MARKET PREDICTION USING ENSEMBLE OF GRAPH THEORY, MACHINE LEARNING AND DEEP LEARNING MODELS

A Project Report

Presented to

Dr. Ching seh Wu

Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Class

CS 298

By

Pratik Patil

May 2019

The Designated Thesis Committee Approves the Thesis Titled

STOCK MARKET PREDICTION USING ENSEMBLE OF GRAPH THEORY, MACHINE LEARNING AND DEEP LEARNING MODELS

by

Pratik Patil

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2019

Dr. Ching seh Wu Department of Computer Science

Dr. Katerina Potika Department of Computer Science

Dr. Marjan Orang Department of Economics

ACKNOWLEDGEMENT

This has been one long and arduous journey, but nevertheless a worthwhile life experience because of the many great Professors at SJSU and beloved friends. I am grateful and take this opportunity to thank my advisor Dr. Wu, who has been my constant support not only during the thesis but during my whole master's degree. It wouldn’t have been possible without his trust and belief in me to do good research.

I also want to thank Dr. Potika and Dr. Orang for consenting to be on my committee and giving their valuable inputs to my project, without which the project would not have been successful.

I would also like to thank my parents, my sister Priya and my beloved friend Shweta for supporting and encouraging me throughout my graduation.

Abstract

Efficient Market Hypothesis (EMH) is the cornerstone of the modern financial theory and it states that it is impossible to predict the price of any stock using any trend, fundamental or technical analysis. Stock trading is one of the most important activities in the world of finance. Stock price prediction has been an age-old problem and many researchers from academia and business have tried to solve it using many techniques ranging from basic statistics to machine learning using relevant information such as news sentiment and historical prices. Even though some studies claim to get prediction accuracy higher than a random guess, they consider nothing but a proper selection of stocks and time interval in the experiments.

In this project, a novel approach is proposed using graph theory. This approach leverages Spatio-temporal relationship information between different stocks by modeling the stock market as a complex network. This graph-based approach is used along with two techniques to create two hybrid models. Two different types of graphs are constructed, one from the correlation of the historical stock prices and the other is a causation-based graph constructed from the financial news mention of that stock over a period. The first hybrid model leverages deep learning convolutional neural networks and the second model leverages a traditional machine learning approach. These models are compared along with other statistical models and the advantages and disadvantages of graph-based models are discussed. Our experiments conclude that both graph-based approaches perform better than the traditional approaches since they leverage structural information while building the prediction model.

Index Terms **- Stock market, machine learning, deep learning, graph theory, financial networks, time series forecasting, spatio-temporal**

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

I. Introduction

A stock is a share or ownership of a part of a publicly listed company. These shares are issued by the company to exchange for traders to trade. These stocks are sold by the owners of the company to raise money/funding for further development of the company. When the company is first listed on an exchange, it is called an Initial Public Offering where the initial selling price of that stock is set by the owners. The price of the stock after the initial public offering is decided by the equilibrium of buy and sell orders, which can also be thought of as demand and supply equilibrium. For example, if there are more people willing to buy a stock than there are who want to sell, then the stock price goes up because of the relatively more demand. However, if there are more people who want to sell the stock than there are who want to buy, then the price drops as demand is less and supply is more.

It is easy to understand the demand and supply is the root cause of price change, however, demand and supply are based on several variables and factors like inflation, positive or negative news, market sentiment, socio-economic factors, trends, and many more.

Since the beginning of the stock market, the goal of the speculators/investors has been to predict the price of the stock as to buy low and sell high, thus earning a profit. For the purpose of this project, stock prices have been assumed to be a time series of equal intervals and different models have been proposed to forecast time series.

A time series has huge significance in Econometrics, Social Science, Healthcare, Cyber Security and can be defined as a sequence of observations collected over regular time intervals. A

time series describes the behavior and change in characteristics of a process over time. If we can understand the process with the help of statistics, its description or graphical representation, then we can model it and use it to forecast the future behavior of the process.

For this project, we will limit our scope of study to time series in econometrics, especially company stock prices.

Since the price of the stock is dependent on a huge number of factors like socio-ecological and sentimental factors, there is a huge amount of noise in the time series which makes it extremely difficult to model using any of the statistical methods.

According to the hypothesis in random walk theory, the prices of a stock market are defined randomly and therefore are impossible to forecast. However, there have been extensive advances in the statistical modeling, deep learning methods, and the availability of huge amount of news and sentimental data. These factors have increased the probability of predicting the stock prices than that of a random process.

There are 3 approaches which can be used to predict stock price.

1) Technical Analysis

2) Fundamental of a company

3) Sentiments of the market

The first technical analysis approach assumes that the price can be predicted from its historical prices and by understanding the patterns like trend and seasonality in the historical time series. In scientific terms, technical analysis is like modeling the stochastic process using which the time series was created.

In the second approach, which is fundamental approach, this is based on the company's financial reports, quarterly earnings, earnings per share, valuation, and other economic factors.

The third approach is the sentiment of the market. Sentiments are what people perceive or believe about the company and its stock price. Sentiments are affected by news. If there is good news favoring the environments/factors needed for a specific company, then people believe that other people will buy the stock and this hysteria affects the price of that stock positively. For example, in 2018, within a day the stock price of Facebook fell by 20% because of the news of Data breach of its users' accounts.

Therefore, we can use machine learning models to predict using the 1st approach of technical analysis and consequently, the 3rd approach of market sentiment can be solved using Natural Language Processing (NLP) on publicly available news.

II. **Research Objective**

The objective of this research is to explore different useful features which can be extracted from graph structural behavior to use in machine learning prediction engine. The project aims to combine graph theory and machine learning by creating a hybrid model for prediction. The aim is to create a generic framework for any time series prediction and not just related to stock prediction.

Chapter 2: Literature Review

I. **Introduction**

Stock market is a complex global market that is open to everyone. Owing to this fact, there are huge number of variables and factors affecting the price of a stock at any given point in time. Therefore, it is almost impossible to predict the future price of a stock and this theory has been proved by Efficient Market Hypothesis (EMH) [32]. However, with the advent of internet and high computational power of GPUs, it has become easy to process high volume of data and find patterns. Thus, a lot of have tried solving the stock price forecasting problem using many different techniques such as Machine learning, Deep learning, and simple statistics. Some researchers have explored the use of news for foresting because of easy access to huge volume of news data available on the internet.

This literature review focusses on time series forecasting, specifically for stock prices. The articles selected for this literature review include conference proceedings, journals, thesis dissertations, and recent research by companies in the field of quantitative analysis of stock market.

The rest of the literature review is organized as follows: Section II presents an overview of the studies done using traditional machine learning algorithms; section III explains existing studies done using graph theory. Section IV explains the recent studies done using deep learning approach. Figure 1 shows the map of the organization of the literature review.

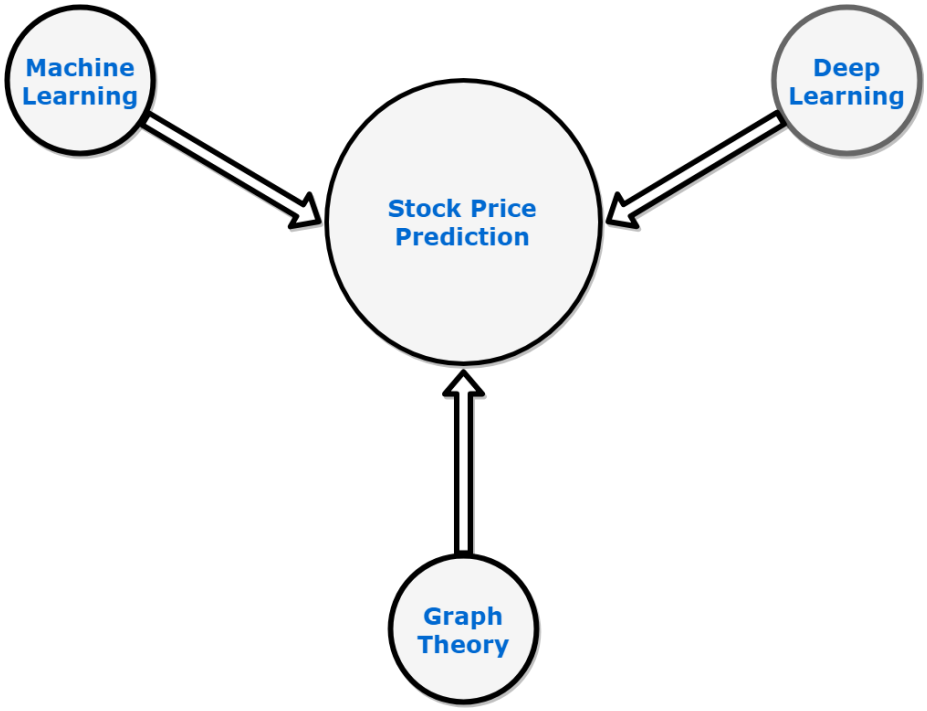


Figure 1: Map of the organization of the literature review

II. **Machine Learning**

Machine learning has been used widely for forecasting stock price. The focus of majority of the studies was to predict change in price for near-term (less than a minute), short-term (1 day ahead or more) and long term (a few months ahead).

Most studies have considered a subset of stocks limited to less than 10 for their study. The set of predictor variables used in the study ranges from simple time series data of stock, google trend, news sentiment data, to characteristics of the company.

Based on the existing work, most of the researchers focus on the near-term predication [1] and long-term [2].

Most of the studies use multiple predictor variables derived from the time series data. These variables are called as indicator in financial terms. See [3] for a well-compiled list. Detailed explanation for every indicator can be found in [4,5]

The most commonly used indicators are listed below in fig 2. [4,5,6]

APO-SMA Absolute price oscillator values with SMA	$PO = \text{SlowMA}(price) - \text{FastMA}(price)$	Moving Average Convergence Divergence (**MACD**)	$shortema = 0.15 \times price + 0.85 * shortema_{[-1]}$ $longema = 0.075 \times price + 0.925 * longema_{[-1]}$ $MACD = shortema - longema$
APO-EMA		Stochastic oscillator (**STOCH**)	$\%K = 100 \times \frac{close - LowestLow_{[\text{last n periods}]}}{HighestHigh_{[\text{last n periods}]} - LowestLow_{[\text{last n periods}]}}$ $\%D = MovingAverage(\%K)$

CCI Commodity channel index values	$CCI = \frac{TP - ATP}{0.015 \times MD}$
	$TP = \frac{high_n + low_n + close}{3}$
	TP = Typical Price
	$high_n$ = Highest high in the last n time periods
	low_n = Lowest low in the last n time periods
	$ATV = \text{SimpleMovingAverage}(TV)$
	$MDTV = \text{MeanDeviation}(TV)$
		Relative strength index (**RSI**) values	If $close > close_{[-1]}$ then
			$up = close - close_{[-1]}$
			$dn = 0$
			else
			$up = 0$
			$dn = close_{[-1]} - close$
			$upavg = \frac{upavg \times (n-1) + up}{n}$
			$dnavg = \frac{dnavg \times (n-1) + dn}{n}$
			$RMI = 100 \times \frac{upavg}{upavg + dnavg}$

AROON	$AroonUp = 100 \times \left(\frac{n - PeriodsSinceHighestHigh}{n}\right)$ $AroonDown = 100 \times \left(\frac{n - PeriodsSinceLowestLow}{n}\right)$		

Bollinger bands (**BBANDS**) values	$TP = \frac{high + low + close}{3}$
	$MidBand = \text{SimpleMovingAverage}(TP)$
	$UpperBand = MidBand + F \times \sigma(TP)$
	$LowerBand = MidBand - F \times \sigma(TP)$
		On balance volume (**OBV**) values.	If $close > close_{[-1]}$ then
			$OBV = OBV_{[-1]} + volume$
			elseIf $close < close_{[-1]}$ then
			$OBV = OBV_{[-1]} - volume$
			else
			$OBV = OBV_{[-1]}$

Chaikin A/D line (**AD**) values.	$CLV = \left(\frac{(close - low) - (high - close)}{(high - low)}\right)$ $AD = AD_{[-1]} + CLV \times volume$		
Average directional movement index (**ADX**) values.	$ADX = \frac{ADX_{[-1]} \times (n-1) + DX}{n}$		

Figure 2: Indicators derived from time series [6]

Supervised Machine Learning algorithms have been used along with the above indicators as features. In addition to indicators some studies use news data as input features to the model. Most of the studies have compared different machine learning algorithm using the same feature set [6]. The study by Alice et. al. compares logistic regression, support vector machine (SVM), and bayesian network and the conclude that SVM are better for smaller time series data [6]. Kim Kj et. al. have compared SVM and artificial neural network (ANN) and they have concluded that SVM are superior for stock price prediction [7]. Study by Huang W. et. al. treated prediction as classification problem and compared linear discriminant analysis, elman back propagation, SVM and quadratic discriminant [8]. Ni LP et. al. used fractal feature selection method along with SVM

and achieved good results [9]. Kumar D et. al. have used 4 different feature selection methods and trained an SVM model for each along with a hybrid model, they concluded that a hybrid model of 4 SMV and random forest performs best [10]. Qiu M et. al. have compared genetic algorithm and ANN after training on a 72 input features gathered form micro and macho economics [11].

Studies have been done on the usage of google trend and number of google searches for a specific stock as input variables to the model. [12, 13].

They conclude that the increase in number of searches implies that the prices will increase within two weeks and will go down within the year. The study by H.J. Kim et. al. used SVM and ANN using google trends, and indicators using in [15] and tested three hypothesis that proved that google trends, and the state machine learning algorithms does not perform well in plausible framework for market investment [14].

III. **Graph Theory Approach.**

It has been proposed by researchers that there exists an underlying structure in the stock market that can be used to understand the behavior of stock markets. In [1], Susan George and Manoj Changat asserted that such underlying structure can be used by governments to prevent a financial crisis such as the one experienced in 2008. They constructed a graph based on the stock market trend correlations and simply calculated various statistics such as the degrees of the nodes. They argue that by simply looking at the resulting graph, various insights can be derived. For example, they discovered that many banks have high degrees, and argued that nodes with a high number of edges are crucial in preventing failures in the stock market. Additionally, these authors argued that the topology of the resulting graphs also describes the market. For example, based on their analysis, a chain-like topology means that there are no powerful companies with high market capitalization, whereas a star-like topology indicates the existence of a few very powerful corporations that have a strong influence over the market. The future work proposed by [1] was to apply community detection algorithms to the graphs constructed from the stock market data;

A very few researchers have tried to model the stock market problem as a graph problem. However, no researchers have significant results so far. Since the stock market does not have inherent or obvious graph structure, it becomes difficult to create a graph representation of stock market.

Researchers have used correlation analysis to construct a graph of stock market, where nodes in the graph are stocks and edges represent the price fluctuation relationship between those stocks [13].

A few scholars have used planar maximally filtered graphs (PMFGs), minimum spanning tree and correlation threshold method for graph construction. Magenta was the first researcher to use

correlation threshold for stock market graph construction [13]. Using this graph, he took the node and important connections between the nodes and created a tree structure to study the hierarchy of the network. This was done using the minimum spanning tree (MST). MST was also used by Kim et al for studying the hierarchy and topology of the stock market structure. [14-17]. Tumminello et. al. used the descriptive statistical characteristics of the stock to calculate the PMFGs. The most promising characteristic he used is the average path length [18]. PMFGs are based on MST but they have more important information than the tree graphs [18]. Papers from [19-23] have used correlation threshold for the construction of the stock market graph. This method for graph construction is the most widely used and most of the recent research done is using the correlation method. The above papers use pearson correlation coefficient with a certain threshold for defining relationship between two stocks. However, XuWeichao et. al. mention that pearson correlation coefficient is linear and thus cannot model non-linear relationships [19]. Therefore they have concluded that spearman rank correlation coefficient and Kendall's Tau are most relevant when modelling non-linear data. Stock price is a highly non-linear time series data and thus [20] have used a spearman rank correlation for their graph construction. Galazka et. al. researched on the Poland stock market by studying a weighted graph using MST [21]. The study concluded that the stock market is scale free. C. K. Tse et. al. studied all the US stocks and concluded that US equity market is scale free [22].

Guangxi et. al. has constructed a correlation threshold-based graph using international stock market [20].

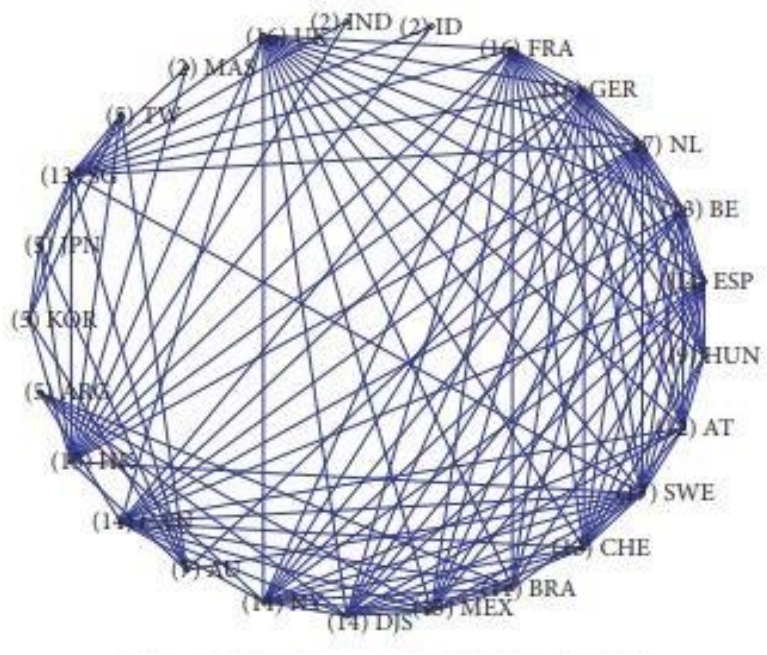


Figure 3: Topology graph of ISMN at threshold 0.4 [20]

In the above figure a graph is constructed with correlation threshold of 0.4 [20]. After creating the different graphs using increasing levels of threshold, they have studied various statistics inside a graph like degree distribution, average path length, and clustering coefficient. As the correlation threshold increases, the degree distribution of the nodes increases, implying that the graph becomes increasingly dense. They randomly removed the edges, nodes and studied the change in the topology of the graph. They concluded that intentional attacks to the network like bankruptcy of a stock or organization breaks the connection integrity of the international stock market network and therefore node with highest degree has the highest effect on overall network [20].

Tharis T. P. et. al have constructed stock market graph from social media posts. They have used the presence of stock name in tweeter posts to construct a relationship. Another dynamic graph was constructed form Kendall correlation coefficient. They have used both these networks to

investigate if the structure of financial market can be determined using past financial information and past information from social media (tweeter). For the purpose of network structure prediction, this study was formulated as a link prediction problem. The study concluded that the financial market structure can be predicted with high accuracy when social media opinion is taken into consideration.[23]

Leandro Anghinoni et. al have developed a trend detection algorithm for stock price using graphs. [24]. They are the first to create a graph where every node in a graph is a state transition and relationship is defined based on the change in trend of stocks. This study used the concepts from the auto regressive integrated moving average (ARIMA) model and used them to create a non-linear framework. The states are created by grouping time series observations in a pre-defined range and then replacing it with a symbol. For instance, let t be the time series, t = [0.1 0.4, 0.6, 0.3, 0.9, 0.5]. With a discretization range of 0.2 we have converted the series to t = [a, b, c, b, f, c].

Leandro Anghinoni et. al explore community detection on stock market graphs. Communities in a graph exhibit an interesting characteristic that groups nodes with similar properties and behavior together. Example of communities is shown in the fig 4. [25]. A community detection algorithm was applied on this state transition graph to form a trend detection algorithm. Their experiments on stochastic time series presents promising results for trend forecasting in multidimensional network.

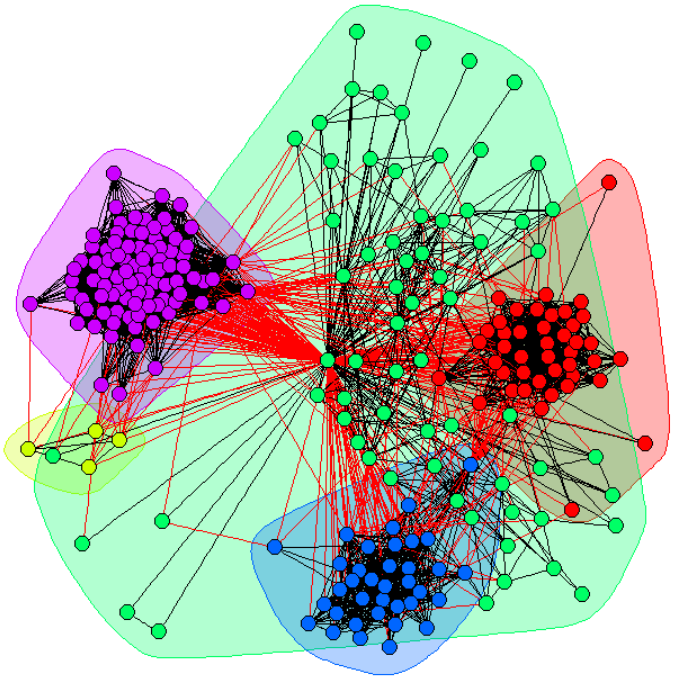


Figure 4: Communities in a graph. [37]

The above figure shows an example of communities within a graph structure. Recently many techniques for community detection in a graph have been developed. Community detection algorithm can be classified into decisive (top down) and agglomerative (bottom-up). In the decisive technique, initially whole graph is a community and depending on some metric nodes are removed to achieve dense communities. The agglomerative technique initially considers every node as community and iteratively merges the communities considering some community metric. One such algorithm is described below.

Louvain Algorithm

Louvain is a greedy optimization algorithm. It aims to optimize modularity. Modularity is a measure of how dense the connection between nodes in a graph is. Local choices are done greedily such that those choices increase the modularity [27]. Modularity is a number between 0-1 [26]. As the goal of Louvain algorithm is to detect communities and clusters, maximum modularity which implies a dense subgraph is needed. A modularity greater than 0.3 is empirically considered a good, which implies presence of good community structure within the graph [27].

Below is the equation that defines the modularity used in the Louvain algorithm [27]:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \qquad (1)$$

Where,

- A_{ij} = The weight of the edge between node i and j
- m = Sum of the weights of all edges, given $(½)*\Sigma_{ij} A_{ij}$
- k_i = Sum of edges incident on node i, given by $\Sigma_j A_{ij}$
- c_i = Community to which node i is assigned
- δ (c_i, c_j) delta function which evaluates to 1 if $c_i = c_j$ (that is, node i and j are in the same community), and evaluates to 0 otherwise (if i and j are not in the same community)

Louvain is a bottom-up algorithm, meaning it start with every node as community and builds its communities from that by adding nodes to communities. The average runtime of this algorithm is O (n log n), where n is the number of nodes in the graph [27].

III. Deep Learning.

Recently a lot of work has been done using deep learning techniques such as recurrent neural networks. Especially long short-term memory (LSTM) are excellent at modelling sequential time series data. De Mello Assis et. al. has trained 30 models and studied the performance of ensemble model for predicting stock price on Brazilian index [28]. Using hypothesis testing they proved that top 5 ANN had an accuracy above 50% [28].

Zhang et. al. have used multiple source as input features, especially financial news and articles [29]. They show that investor opinions through news and social media has significant effect on the market volatility. Sentiment analysis was done on financial news and fed as an input to the model. They concluded that accurate consistent news information significantly increases the accuracy of the model. Jiahong li et al have used LSTM with news sentiment analysis and achieved an accuracy of 88%.

Recently an emerging branch of deep learning known as reinforcement learning (RL) have showed promising opportunities in the field of stock market prediction. Reinforcement learning agent is based on a certain set of actions and a goal which in this case is to maximize the profit [31]. A deep recurrent Q-learning was employed in this study and the results of the experiment were positive profit. This is the first positive results by a pure deep reinforcement learning algorithm under transactional costs and therefore RL provides promising opportunities for researchers in this field.

Chapter 3: Implementation Platform and Libraries Used

I. Data and Graph Manipulation Libraries:

Programming language Python 3 has been utilized to implement the complete project. This is the language of choice for the majority of the data science project because of its support for machine learning and data analysis. Other libraries that were used in the project are Tensorflow, pandas, numPy, sciPy, pyramid for Auto regressive integrated moving average (ARIMA), networkX, scikit-learn, matplotLib, fix_yahoo_finance, ta.

Pandas is used for efficient and compatible data storage in python, it also has in-built support for extensive data manipulations operations. NumPy provides scientific computation and vector manipulations from the CSV data format. It is also compatible with all the major ML libraries in python. Tensorflow is a library to implement and simulating deep learning models. This is required to implement convolution layers in GCN. Tensorflow leverages the compute capacity of GPUs in addition to CPU. NetworkX library is used for all the graph creation, visualization, and manipulation. It has in-built support for many graph algorithms like community detection and page rank. Ta is a time series analysis library which creates new features from input time series. Scikit-learn is a library which has an end to end pipeline for all machine learning algorithms. Fix_yahoo_finance was used to scrape stock price data from yahoo finance website.

II. Development Platform:

a) Setup on local computer (Windows 10)

A local desktop setup was used in the initial stages of the project which involved data pre-processing and graph creation. This computer has i7 processor with 16 GB RAM and 940M Nvidia

GeForce GPU. Libraries were used from anaconda environment which has a collection of all the libraries required for data science and machine learning. The steps and instructions to setup anaconda can be found on https://anaconda.org.

b) Collab by Google.

Collab is a cloud service by google which supports GPU accelerated deep learning on jupyter notebook. The cloud platform is perfect for deep learning application as it comes pre-loaded with libraries like tensorflow, pandas, keras, and pytorch. However, there are certain restrictions on session time and size of the file uploaded. But certainly, it is a great tool for developing deep learning algorithms leveraging GPUs. The steps and instructions for google collab can be found at https://colab.research.google.com/

c) Amazon Web Services (AWS)

The project was later moved to AWS owing to the limitations of google collab. AWS has a pre-built dockerized image called Amazon Machine Image (AMI) which has all the machine learning platforms and libraries on it. For the storage S3 bucket was utilized. AMI comes with support for GPUs. The steps and instruction to setup AMI platform can be found at https://aws.amazon.com/blogs/machine-learning/get-started-with-deep-learning-using-the-aws-deep-learning-ami/

Chapter 4: Dataset

The raw data used in this project is the time series data, that is the stock price data collected at an equal interval of time. This stock prices data was collected for 30 stocks from the top 30 fortune 500 companies listed below in table 3. This dataset was collected for two different intervals, 1-day interval, and 1 min interval.

Another dataset that is used in this project is news dataset. This data contains 1 million financial news collected from a financial news website.

Table 1: List of 30 stocks using in the project

Sr. No	Ticker	Company Name	Sector
1	WMT	Walmart	Consumer
2	XOM	Exxon Mobil Corporation	Energy
3	AAPL	Apple Inc.	Technology
4	UNH	UnitedHealth Group Incorporated (DE)	Healthcare
5	MCK	McKesson Corporation	Healthcare
6	CVS	CVS Health Corporation	Healthcare
7	AMZN	Amazon.com Inc	Technology
8	T	AT&T Inc	Communication Services
9	GM	General Motors Company	Auto Mobile
10	F	Ford Motor Company	Auto Mobile
11	ABC	AmerisourceBergen Corporation	Healthcare
12	CVX	Chevron Corporation	Energy
13	CAH	Cardinal Health Inc	Healthcare
14	COST	Costco Wholesale Corporation	Consumer Defensive

15	VZ	Verizon Communications Inc	Communication Services
16	KR	Kroger Company	Consumer
17	GE	General Electric Company	Industrials
18	WBA	Walgreens Boots Alliance Inc	Healthcare
19	JPM	JP Morgan Chase & Co	Financial Services
20	GOOGL	Alphabet Inc	Technology
21	HD	Home Depot Inc	Consumer
22	BAC	Bank of America Corporation	Financial Services
23	WFC	Wells Fargo & Company	Financial Services
24	BA	The Boeing Company	Industrials
25	PSX	Phillips 66	Energy
26	ANTM	Anthem Inc	Healthcare
27	MSFT	Microsoft Corporation	Technology
28	UNP	Union Pacific Corporation	Industrials
29	PCAR	PACCAR Inc	Industrials
30	DWDP	DowDuPont Inc	Basic Materials

I. Stock Price Data Collection:

a) 1-Day interval dataset:

Daily closing prices of these 30 stocks were collected by scrapping the website https://finance.yahoo.com. Two years of data was collected starting from 2017-05-01 to 2019-04-01. Fix_yahoo_finance was used in scrapping the daily closing price of each stock. This data was only used in creating the graph of companies and calculating the correlation matrix because it has more variance than minute level data and thus is able to model richer information. It was not used for training and predicting the GCN model.

b) 1-minute interval dataset.

Minute level data of stock prices are not publicly available since it requires substantial storage. However, there are specific financial data companies which store this data. This data is exposed to data scientists using a restful API for a small amount of fee.

For the purpose of this project, minute-level data for above 30 stocks were collected form the website https://api.tiingo.com. This data is later used in graph convolution networks (GCN) and linear machine learning (ML) models for training and forecasting.

The stock market is open for trading for 6.5 hours daily from Monday to Friday, which is 390 minutes daily. Therefore, everyday contained 390 data points. Data was collected for 44 days for all 30 stocks. Finally, every time series (stock) had 17,204 data points.

II. News dataset:

1 million financial news were collected from a financial news website. The data was collected since the year 2007. The dataset lists the news and the companies mentioned in that particular piece of news. For example, in the figure below, the news with the headline "PRESS DIGEST- New York Times -Jan 1" mentioned 6 different companies (Google, XM Radio, Sirius XM, Walt Disney, Microsoft, and Yahoo). This data was later used to generate the causality-based graph.

sourceTim	firstCreate	sourceId	headline	assetCode	assetName	firstMenti
2007-01-(	2007-01-(	e58c6279	China's Daqing pumps 43.41 mln tonnes of oil in 06	{'PTR.N', '(	PetroChina Co Ltd	6
2007-01-(	2007-01-(	5a31c432	FEATURE-In kidnapping, finesse works best	{'STA.N'}	Travelers Companies Inc	8
2007-01-(	2007-01-(	1cefd27a4	PRESS DIGEST - Wall Street Journal - Jan 1	{'WMT.N',	Wal-Mart Stores Inc	14
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'GOOG.O	Google Inc	13
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'XMSR.O'	XM Satellite Radio Holdings Inc	11
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'SIRI.OQ',	Sirius XM Radio Inc	0
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'DIS.N', '[	Walt Disney Co	5
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'MSFT.DE	Microsoft Corp	10
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'YHOO.O'	Yahoo! Inc	13
2007-01-(	2007-01-(	9fb959be4	Tenet Completes Sale of Alvarado Hospital Medical Center <THC	{'THC.N'}	Tenet Healthcare Corp	1
2007-01-(	2007-01-(	abf2fa5ad	RPT-Wall St Week Ahead: Mild jobs may lift stocks as '07 starts	{'SIEB.OQ'	Siebert Financial Corp	3
2007-01-(	2007-01-(	abf2fa5ad	RPT-Wall St Week Ahead: Mild jobs may lift stocks as '07 starts	{'MON.N',	Monsanto Co	0
2007-01-(	2007-01-(	abf2fa5ad	RPT-Wall St Week Ahead: Mild jobs may lift stocks as '07 starts	{'STZ.N', 'S	Constellation Brands Inc	40
2007-01-(	2007-01-(	3892aac4	FEATURE-In kidnapping, finesse works best	{'STA.N'}	Travelers Companies Inc	8
2007-01-(	2007-01-(	107f4407	Rite Aid Can Help You Realize Your New Year's Weight Loss Reso	{'RAD.N'}	Rite Aid Corp	1
2007-01-(	2007-01-(	24312f29	Commtouch Reports: "Happy New Year!" Virus Ends 2006 with	{'CTCH.O'}	Cyren Ltd	1
2007-01-(	2007-01-(	77f06424	Seoul antitrust body forms team on Qualcomm-report	{'MSFT.DE	Microsoft Corp	11
2007-01-(	2007-01-(	77f06424	Seoul antitrust body forms team on Qualcomm-report	{'QCOM.O	Qualcomm Inc	1
2007-01-(	2007-01-(	1ff5d3791	Talks to save BenQ Mobile Germany fail	{'SI.N', 'SI	Siemens AG	6

Figure 5: Excerpt from the new dataset

Chapter 5: Modelling stocks into a graph.

The goal of this project is to model the stock price prediction problem into a graph problem. Here, to create a stock graph, we can assume that the stocks are nodes of a graph. However, the most important question is how to define the relationship between two nodes/stocks.

In this project, the relationship between any two nodes is defined in two ways:

1) Correlation based relationship.
2) Causation based relationship.

I. Correlation based relationship.

The correlation coefficient is a numerical measure of the statistical relationship between two variables. This number is between -1 and 1 representing linear dependence between two variables. The correlation coefficient is calculated between two columns of data. If both variables rise and fall together then the correlation coefficient is positive with the higher magnitude and vice versa.

To define the relationship between two nodes. We calculate a correlation coefficient between two stocks and if it is greater than a certain threshold then we create an edge between those two nodes.

To make this easier, correlation matrix is calculated and if the absolute value of correlation coefficient of stock I, j >threshold (0.5), then we create an edge between stock I and j by putting 1 at location matrix[I,j] other we put 0. Eventually, we get an adjacency matrix which can be interpreted as a graph. The graph structure changes as the threshold vary from 0-1.

In this project, we have used the following 3 correlation coefficient to generate the graph.

1) Pearson correlation coefficient
2) Spearman correlation coefficient
3) Kendall correlation coefficient

All the 3 correlation coefficients are calculated on the top 30 stocks from fortune 500 mentioned above. Every stock price is a daily closing price with 1-day interval.

a) Spurious correlation problem:

The correlation coefficient is calculated between two series. However, for time series, the correlation is calculated at different lags which is often called a cross-correlation function.

Pearson correlation gives us the “linear” correlation between two sequences. Pearson correlation is appropriate for independent data only. The usage of the correlation coefficient in the context of time series is specific, where we need to take into account the within series dependency, otherwise, the correlation will be spurious. If we want to use the correlation coefficient on this kind of data, then the data should be first differenced and calculate the correlation coefficient on the increments. [32].

The correlation coefficient is also sensitive to variation. Therefore, if two time series are strongly dependent without any within series dependence, however, one time series has high variance and other has low variance even then the correlation coefficient will be low and spurious.

To avoid these both scenarios, we do the following:

Differencing: $X_t = S_t - S_{t-1}$

To Remove Variance: log(X)

b) Pearson correlation coefficient

Pearson correlation measures the linear relationship between given two input series. The below figures from 7-10 are the graphs constructed using pearson correlation coefficient with different thresholds. The formula for calculating Pearson coefficient is:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\ \sqrt{n\sum y_i^2 - (\sum y_i)^2}}.$$

Figure 6: Pearson Correlation Coefficient [33]

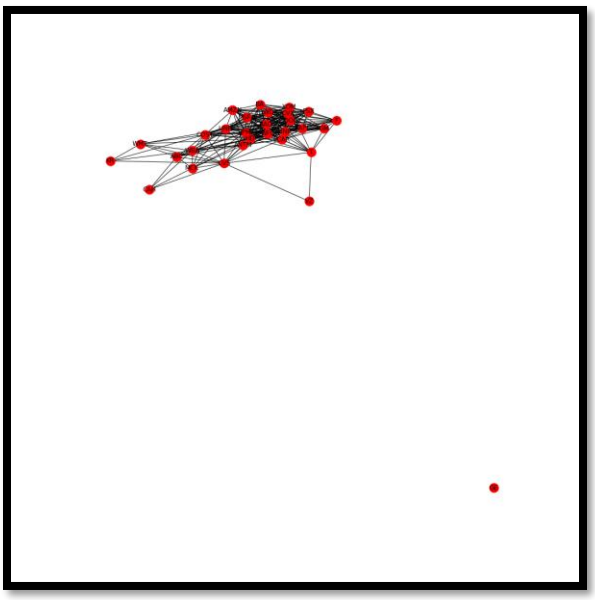


Figure 7: Graph from Pearson coef threshold of 0.3

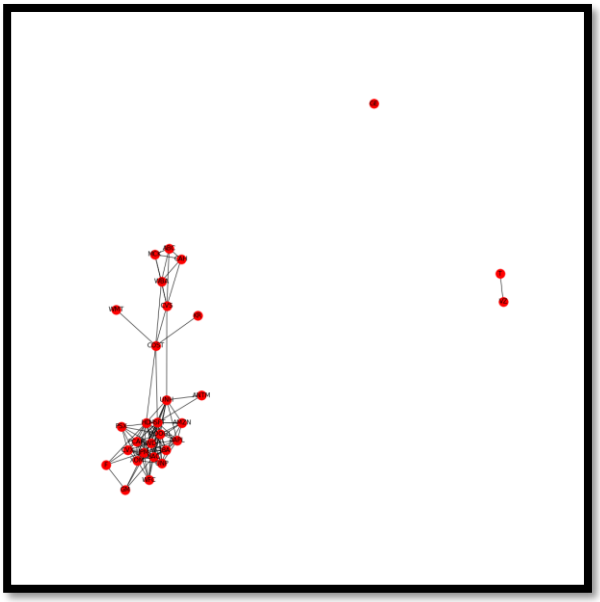


Figure 8: Graph from Pearson coef threshold of 0.4

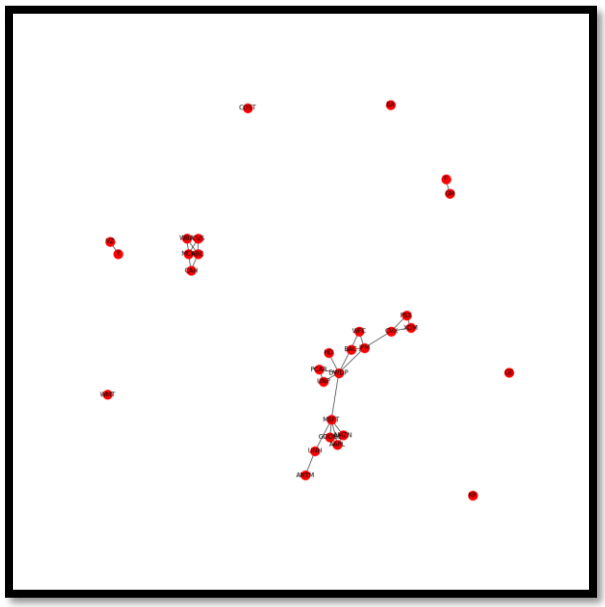


Figure 9: Graph from Pearson coef threshold of 0.5

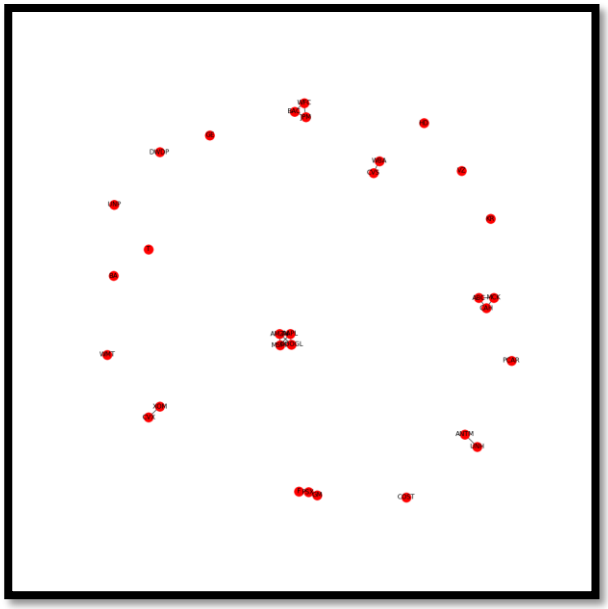


Figure 10: Graph from Pearson coef threshold of 0.6

c) Spearman correlation coefficient

Spearman correlation measures the non-linear relationship between given two input series. This is more robust as compared to Pearson correlation. The below figures from 12-15 are the graph constructed using spearman correlation coefficient with different thresholds. The formula for calculating Spearman coefficient is:

$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}}$$

Figure 11: Spearman Rank Correlation Coefficient [33]

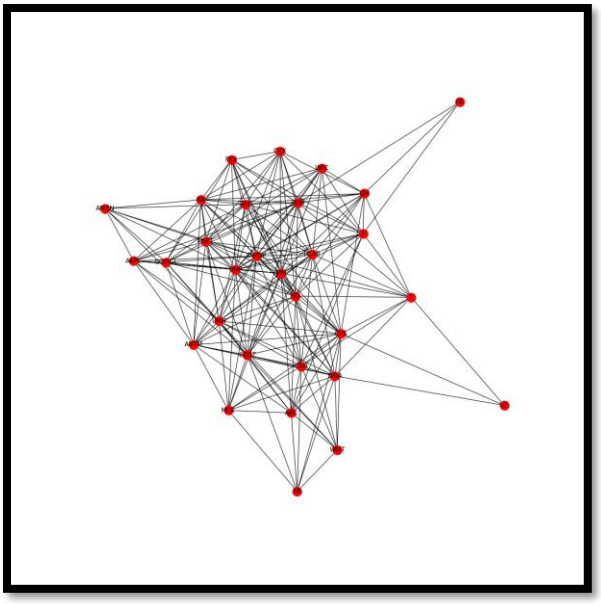


Figure 12: Graph from Spearman coef threshold of 0.3

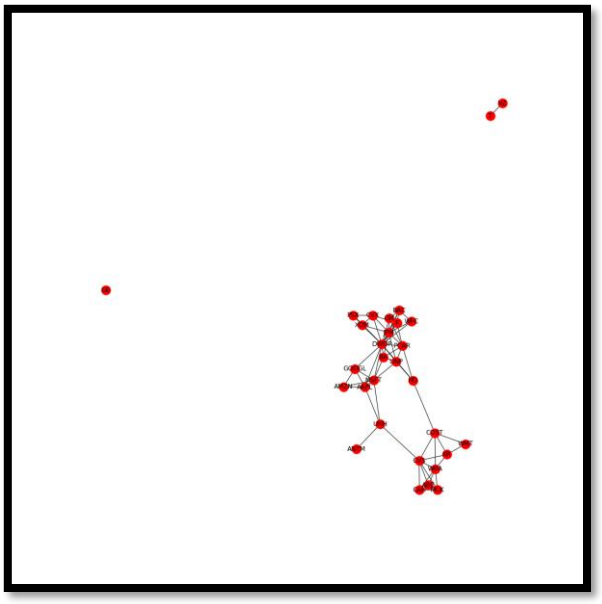


Figure 13: Graph from Spearman coef threshold of 0.4

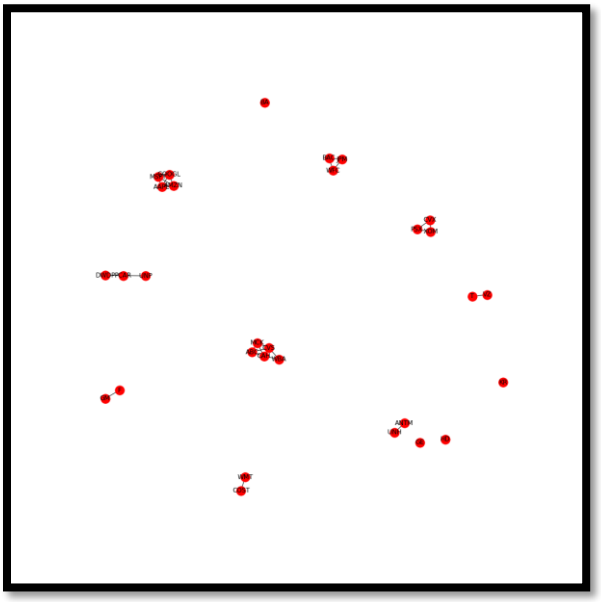


Figure 14: Graph from Spearman coef threshold of 0.5

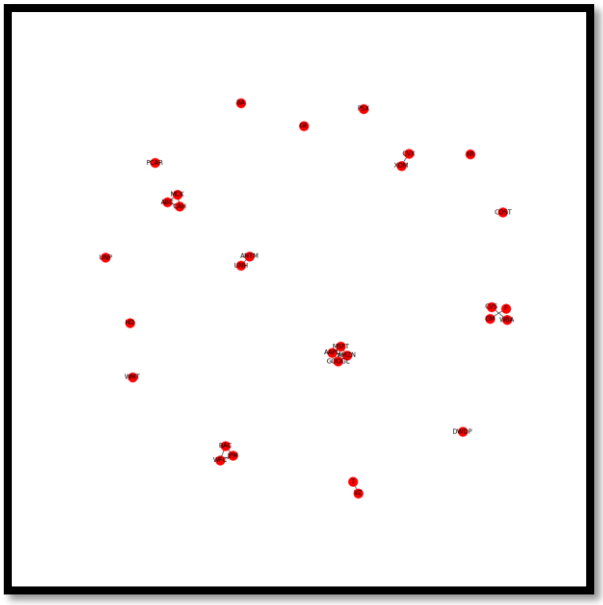


Figure 15: Graph from Spearman coef threshold of 0.6

d) Kendall correlation coefficient

The below figures from 17-20 are the graph constructed using kendall tau correlation coefficient with different thresholds.

$$\tau_b = \frac{S}{\sqrt{\left[n(n-1)/2 - \sum_{i=1}^{t} t_i(t_i-1)/2\right]\left[n(n-1)/2 - \sum_{i=1}^{u} u_i(u_i-1)/2\right]}}$$

Figure 16: Kendall Correlation Coef [33]

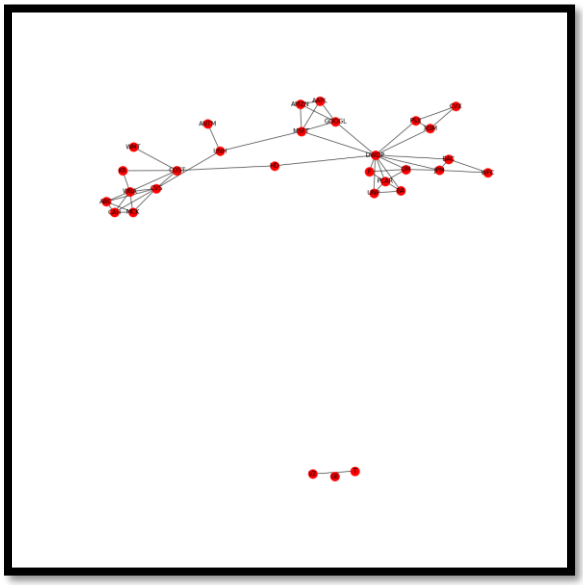


Figure 17: Graph from Kendall coef threshold of 0.3

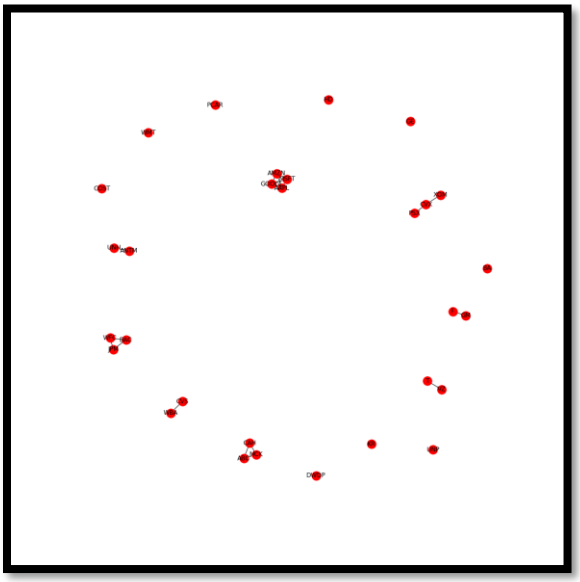
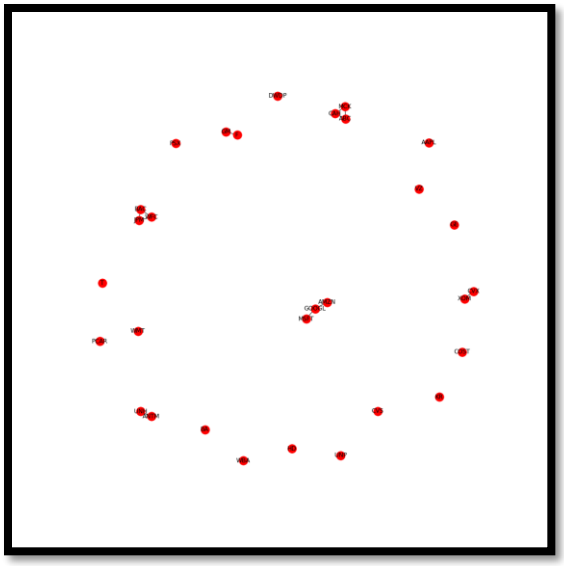


Figure 18: Graph from Kendall coef threshold of 0.4



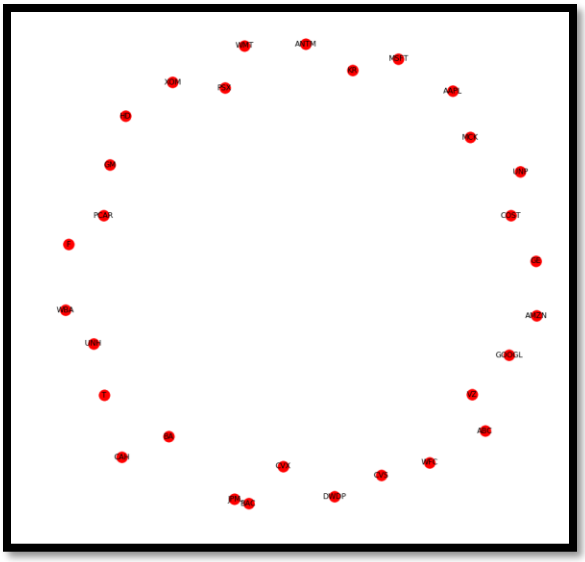


Figure 19: Graph from Kendall coef threshold of 0.5

Figure 20: Graph from Kendall coef threshold of 0.6

e) Choosing the correct threshold.

The graph structure changes as the threshold vary from 0-1. As the threshold increases, the graph becomes more sparse. If the threshold is small, the graph is well connected as the condition for setting an edge between nodes is loose.

To determine the optimal value of threshold which creates a graph with useful information, an experiment was conducted to detect communities within the graphs.

One of the most relevant and useful features of a graph structure is its clustering of nodes or community. A community structure within a graph is defined as a set of nodes which can be grouped together and whose nodes are densely connected internally. In real-world graphs, nodes within a community can be considered to be related in a similar manner.

In the case of a stock graph, communities within this graph should represent a sector or a specific industry like healthcare, finance, technology, etc. This is based on the assumption that the stock

prices of companies within the same sector would rise and fall together because of similar opportunities and conditions for them. Therefore, community detection algorithm was used to select appropriate threshold value in this project.

In this project, the louvain community detection algorithm was used to create communities. Louvain is a modularity-based algorithm. Empirically a modularity value greater than 0.4 represents good community structure in the graph. From the experiments, the threshold of 0.5 gave the most relevant communities for all the 3 correlation types, Pearson, Kendall, and Spearman. The below table lists the communities and its corresponding sector for the threshold of 0.5 calculated by Pearson correlation coefficient.

Table 2: Stocks listed based on communities

Community	Ticker	Company Name	Sector
1	AAPL	Apple Inc.	Technology
1	AMZN	Amazon.com Inc	Technology
1	GOOGL	Alphabet Inc	Technology
1	MSFT	Microsoft Corporation	Technology
2	CVS	CVS Health Corporation	Healthcare
2	ABC	AmerisourceBergen Corporation	Healthcare
2	MCK	McKesson Corporation	Healthcare
2	CAH	Cardinal Health Inc	Healthcare
2	WBA	Walgreens Boots Alliance Inc	Healthcare
3	UNH	UnitedHealth Group Incorporated (DE)	Healthcare

3	ANTM	Anthem Inc	Healthcare
4	BA	The Boeing Company	Industrials
5	BAC	Bank of America Corporation	Financial Services
5	WFC	Wells Fargo & Company	Financial Services
5	JPM	JP Morgan Chase & Co	Financial Services
6	COST	Costco Wholesale Corporation	Consumer Defensive
7	CVX	Chevron Corporation	Energy
7	PSX	Phillips 66	Energy
7	XOM	Exxon Mobil Corporation	Energy
8	UNP	Union Pacific Corporation	Industrials
8	PCAR	PACCAR Inc	Industrials
8	DWDP	DowDuPont Inc	Basic Materials
8	HD	Home Depot Inc	Consumer
9	GM	General Motors Company	Auto Mobile
9	F	Ford Motor Company	Auto Mobile
10	GE	General Electric Company	Industrials
11	KR	Kroger Company	Consumer
12	T	AT&T Inc	Communication Services
12	VZ	Verizon Communications Inc	Communication Services
13	WMT	Walmart	Consumer

II. **News based relationship**.

Financial news were used in generating another type of stock graph. In this graph, the relationship between two nodes/stock is based on co-mention of those two stocks in the same news article. The assumption that two stocks mentioned in the same news must be somehow related to each other was used here.

sourceTim	firstCreate	sourceId	headline	assetCode	assetName	firstMenti
2007-01-(	2007-01-(	e58c6279	China's Daqing pumps 43.41 mln tonnes of oil in 06	{'PTR.N', '(	PetroChina Co Ltd	6
2007-01-(	2007-01-(	5a31c432	FEATURE-In kidnapping, finesse works best	{'STA.N'}	Travelers Companies Inc	8
2007-01-(	2007-01-(	1cefd27a4	PRESS DIGEST - Wall Street Journal - Jan 1	{'WMT.N',	Wal-Mart Stores Inc	14
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'GOOG.O	Google Inc	13
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'XMSR.O'	XM Satellite Radio Holdings Inc	11
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'SIRI.OQ'	Sirius XM Radio Inc	0
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'DIS.N', 'I	Walt Disney Co	5
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'MSFT.DE	Microsoft Corp	10
2007-01-(	2007-01-(	23768af19	PRESS DIGEST - New York Times - Jan 1	{'YHOO.O'	Yahoo! Inc	13
2007-01-(	2007-01-(	9fb959be4	Tenet Completes Sale of Alvarado Hospital Medical Center <THC	{'THC.N'}	Tenet Healthcare Corp	1
2007-01-(	2007-01-(	abf2fa5ad	RPT-Wall St Week Ahead: Mild jobs may lift stocks as '07 starts	{'SIEB.OQ'	Siebert Financial Corp	3
2007-01-(	2007-01-(	abf2fa5ad	RPT-Wall St Week Ahead: Mild jobs may lift stocks as '07 starts	{'MON.N',	Monsanto Co	0
2007-01-(	2007-01-(	abf2fa5ad	RPT-Wall St Week Ahead: Mild jobs may lift stocks as '07 starts	{'STZ.N', 'S	Constellation Brands Inc	40
2007-01-(	2007-01-(	3892aac4	FEATURE-In kidnapping, finesse works best	{'STA.N'}	Travelers Companies Inc	8
2007-01-(	2007-01-(	107f4407(	Rite Aid Can Help You Realize Your New Year's Weight Loss Reso	{'RAD.N'}	Rite Aid Corp	1
2007-01-(	2007-01-(	24312f29	Commtouch Reports: "Happy New Year!" Virus Ends 2006 with	{'CTCH.O'}	Cyren Ltd	1
2007-01-(	2007-01-(	77f06424	Seoul antitrust body forms team on Qualcomm-report	{'MSFT.DE	Microsoft Corp	11
2007-01-(	2007-01-(	77f06424	Seoul antitrust body forms team on Qualcomm-report	{'QCOM.O	Qualcomm Inc	1
2007-01-(	2007-01-(	1ff5d3791	Talks to save BenQ Mobile Germany fail	{'SI.N', 'SI	Siemens AG	6

Figure 21: Co-mention of stocks in new articles

Therefore, if a piece of news mentions Amazon, Microsoft, and Apple in the same article, then we create an edge between each of these 3 nodes/stocks with an edge weight of 1. If another news mentions Apple and Microsoft in the article, and if there is already an edge between these nodes then we increase the edge weight by 1 to represent a stronger correlation. This, when done over a dataset of 1 million news, would remove any outliers or wrong mention of any two companies in the same news article. The figure below is a representative graph constructed from a news article. However, for the purpose of the model building in this project, the graph with only 30 nodes/stocks was considered

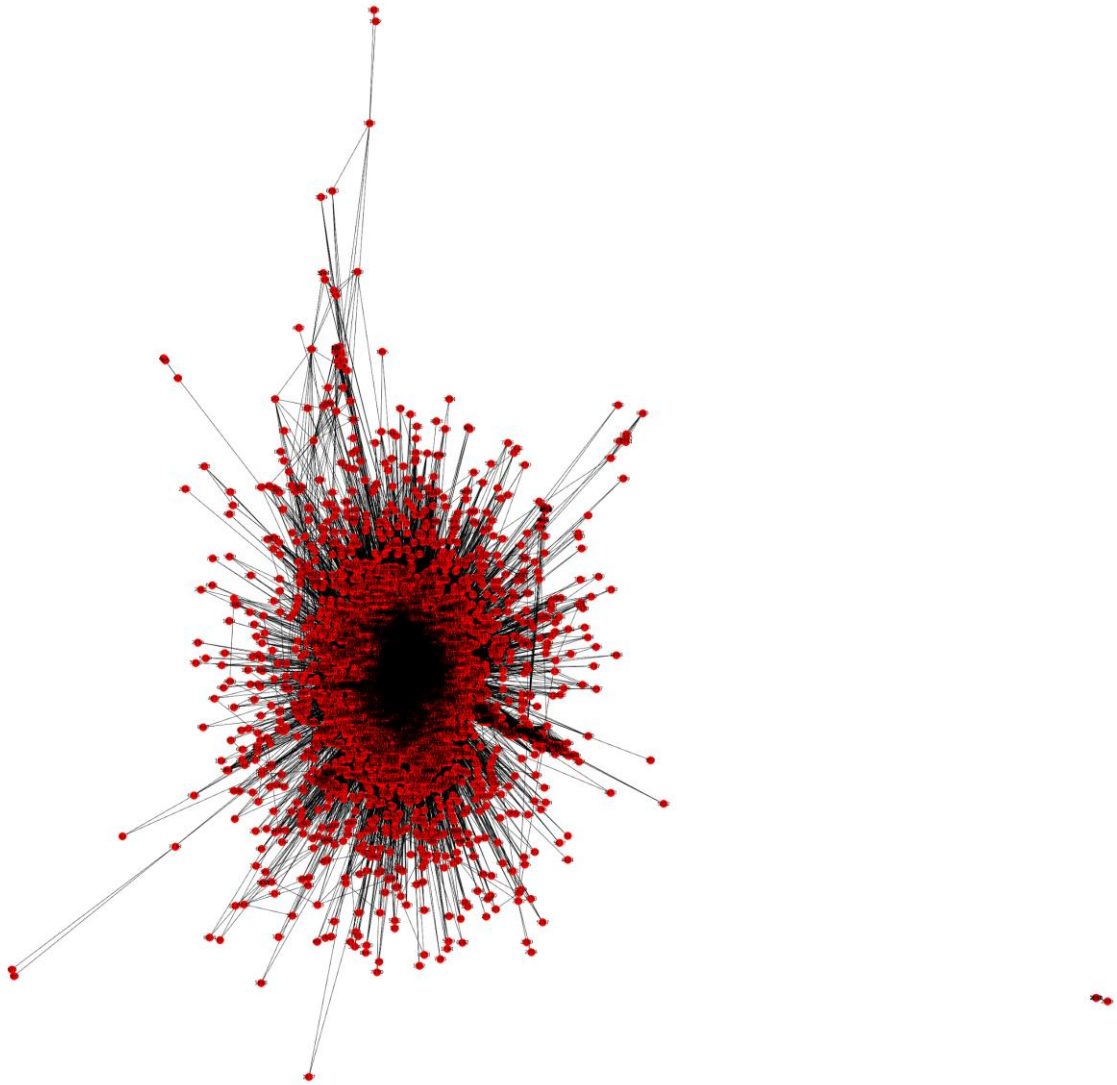


Figure 22: Graph Based on News (Includes stocks on all exchanges)

Chapter 6: Approaches and Implementation.

Time series forecasting has traditionally been done using statistical algorithms like auto regressive integrated moving average (ARIMA), holtwinters and even using LSTMs and recurrent neural unites recently. However, each model needs to be built for every stock/time series and thus all these models are independent of each other. They do not leverage the Spatio-temporal relationship between different companies.

In this project two novel models are proposed based on the graph which leverage the Spatio-temporal relationship between different stocks/companies. The first model is based on deep learning convolutional neural network and the other model is based on a traditional machine learning algorithm.

I. Graph Based Deep Learning Models

In this model, we create a graph convolution neural network (GCN), in which we construct convolution layers based on graph and its structure instead of using regular recurrent units and convolutions. In this model, we create multiple blocks of spatio-temporal convolution which are comprised of graph convolution layers. We extract spatio-temporal features from the graph of many time series (stock prices) using convolution structures.

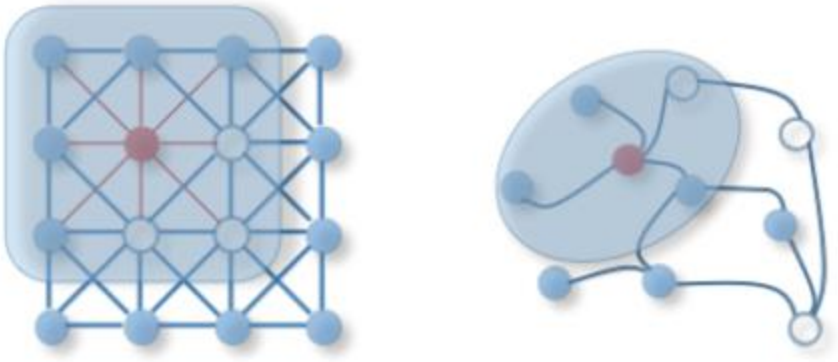


Figure 23: Convolution in 2-d (Image Pixels) and 3-d (Graph) [35]

In the above figure 23, on the left side we have 2d image with pixels and the right side figure is a graph. In 2d convolution, usually an image, we determine the neighbor of a pixel using a filter or window size. In this filter, we calculate the weighted average of the neighboring pixels. Similarly, for a graph convolution, instead of a specific filter, we can fix on level of depth of neighboring nodes. Therefore, to collect features for red node we can all features of neighboring nodes of red node in the graph until a specific depth.

a) Definitions and nomenclature:

In this project, a graph is represented by $\mathcal{G} = (\mathbf{V}, \mathbf{E}, \mathbf{A})$, where V is the nodes in the graph, in this case the stocks/companies. E is the set of edges and A is the adjacency matrix. A is of size nxn and the weight of every edge A[i][j] >0 if there is edge between I and j and A[i][j] = 0 if there is no edge. Every node in the graph has a feature vector represented by X. In this case the feature vector is a temporal time series.

Therefore, a spatio-temporal graph is a graph whose structure is constant but the feature vector X changes over time. The figure below 24 shows a representation of temporal graph over time period.

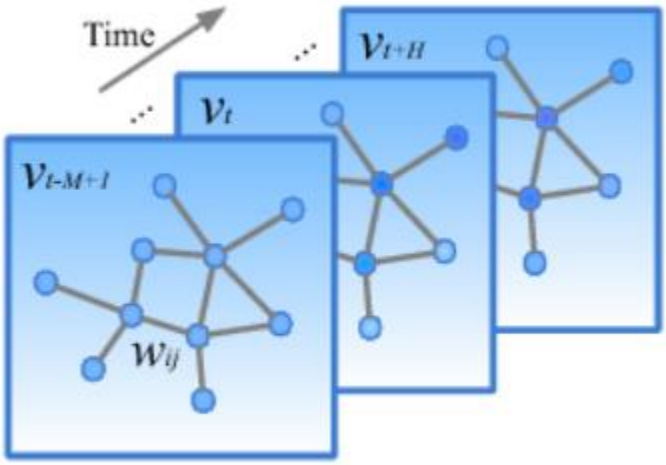


Figure 24: Graphical representation of temporal signal imitating graph signal.[34]

Below figure 25 describes a general architecture for a Spatio-temporal graph convolutional neural network. The adjacency matrix A is passed to the architecture along with the feature matrix X. In this case the feature matrix is a vector containing time series of the corresponding node. Then there is a GCN layer which calculates the spatial dependency from input A and the next 1-D CNN layers [34] calculates the temporal dependency from the feature vector X. There is a Multi layered perceptron (MLP) layer after this which does a linear transformation to predict at each node/stock [36].

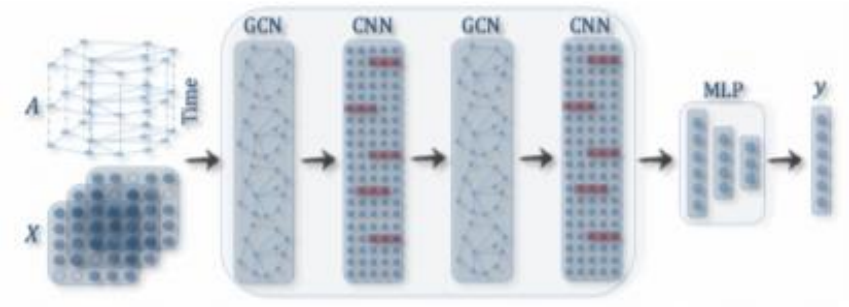


Figure 25: Graph Convolution Neural Network Architecture [36]

There are two inputs to this model:

1) Adjacency matrix of the graph (Weight matrix)
2) Time series (stock prices) of every node in the graph.

The adjacency matrix is calculated using two ways, correlation coefficient and using news co-mentions. For the time series, the interval is set to 1 minute. In the US, the stock market is open for 6.5 hours, which makes the time series to have 390 data points per day. The linear interpolation is used to fill NA or missing values. Normalization is done to scale all the time series within the range 0 and 1. Prediction is done for 3 ,6 and 9-time steps ahead [34]. Since the data is normalized, we calculate the merged Root mean squared error (RMSE), Mean absolute percentage error (MAPE), and mean absolute error (MAE) for model evaluation.

II. Graph Based Traditional ML Models

In this model, create a linear regression model for predicting stock prices. A separate model is needed for predicting every stock. To build a model for a stock we extract the features from that stock's time series. We also derive features from time series of other stocks. The additional stocks which are used in the feature set are determined from the communities in the graph structure.

a) Community Detection:

One of the most relevant and useful features of a graph structure is its clustering of nodes or community. A community structure within a graph is defined as a set of nodes which can be grouped together and whose nodes are densely connected internally. In real world graphs, nodes within a community can be considered to be related in similar manner.

In case of stock graph, communities within this graph should represent a sector or specific industry like healthcare, finance, technology etc. This is based on the assumption that stock prices of companies within same sector would rise and fall together because of similar opportunities and conditions for them. Therefore, community detection algorithm was used to select appropriate threshold value in this project.

In this project, Louvain community detection algorithm was used to create communities. Louvain is a modularity-based algorithm. Empirically a modularity value greater than 0.4 represents good community structure in the graph.

To predict a specific stock, a model is built by training on features extracted from other stocks in the same community.

b) Feature Extraction

22 features are extracted from the time series using statistical formulas. These features are called indicator in technical analysis of stock market. Below is the list of all the technical indicators extracted in this project.

Table 3: List of all the statistical indicator(features) extracted from time series

Sr. No.	Indicator Name	Description
1	Bollinger High Band Indicator	Returns 1, if close is higher than bollinger high band. Else, return 0.
2	Relative Strength Index (RSI)	
3	True strength index (TSI)	
4	Bollinger Bands (BB)	Upper band at K times an N-period standard deviation above the moving average (MA + Kdeviation).
5	Bollinger Low Band Indicator	Returns 1, if close is lower than bollinger low band. Else, return 0.
6	Bollinger Bands (BB)	N-period simple moving average (MA).
7	Donchian channel (DC)	The upper band marks the highest price of an issue for n periods.
8	Donchian High Band Indicator	Returns 1, if close is higher than donchian high band channel. Else, return 0.
9	Donchian channel (DC)	The lower band marks the lowest price for n periods.
10	Donchian Low Band Indicator	Returns 1, if close is lower than donchian low band channel. Else, return 0.
11	Aroon Indicator (AI)	Identify when trends are likely to change direction (downtrend). Aroon Down - ((N - Days Since N-day Low) / N) x 100
12	Aroon Indicator (AI)	Identify when trends are likely to change direction (uptrend).
13	Detrended Price Oscillator (DPO)	Is an indicator designed to remove trend from price and make it easier to identify cycles.
14	EMA	Exponential Moving Average via Pandas

15	Moving Average Convergence Divergence (MACD)	Is a trend-following momentum indicator that shows the relationship between two moving averages of prices.
16	Moving Average Convergence Divergence (MACD Diff)	Shows the relationship between MACD and MACD Signal.
17	Moving Average Convergence Divergence (MACD Signal)	Shows EMA of MACD.
18	Trix (TRIX)	Shows the percent rate of change of a triple exponentially smoothed moving average.
19	Cumulative Return (CR)	Cumulative Return (CR)
20	Daily Log Return (DLR)	
21	Daily Return (DR)	Daily Return (DR)
22	Bollinger Bands (BB)	Lower band at K times an N-period standard deviation below the moving average (MA − Kdeviation).

c) Formulating the time series forecasting problem as supervised machine learning

In this model, linear regression model has been used to predict the future price. To build a linear model and predict the price at time step t+1, we need to have features at time t+1. This is not possible since that is what we are trying to predict.

Therefore, to solve this problem, we shift the label, that is vector Y by an offset of h, where h is the number of time steps we need to forecast in future.

Below in the table 4 we have feature matrix X and the corresponding true labels in vector Y at each time step. In table 5 we have offset the labels vector by 1. This means during model training the label for time step t is value of stock at time step t+1. Therefore, in the predict phase, if the input to the model is features derived at time step t, then the model will output the future price at time step t+1. A separate model needs to be built for every offset value and stock.

Table 4: Input feature matrix and label vector Y

X (Feature matrix)			Y (Label)
1.8	1.5	1.5	1
1.9	2.2	3.2	2
2.4	2.1	2.3	3
4.5	5.3	5.4	4
6.4	2.5	4.5	5
6.2	6.3	7.3	6
6.9	3.3	5.3	7
8.3	8.9	8.5	8

Table 5: Input feature matrix and label vector Y with OFFSET=1 (Forecast 1 time step ahead)

X (Feature matrix)			Y (Label)
1.8	1.5	1.5	
1.9	2.2	3.2	2
2.4	2.1	2.3	3
4.5	5.3	5.4	4
6.4	2.5	4.5	5
6.2	6.3	7.3	6
6.9	3.3	5.3	7
8.3	8.9	8.5	8

d) Building Linear Models

In this part, two different linear models are built for every stock, each with different input feature matrix. We call it single model and a composite model. The single model derives its features from its own time series. The other composite model derives its features from all the neighboring stocks in the same community of the graph. The communities derived using the Louvain algorithm is used in this part. Below in figure 26, we have a list of stocks in each community. For example, we need to build both single and composite model for the stock AAPL (Apple. Inc.).

a) Single model:

22 features mentioned in table 3 are extracted from time series of AAPL for training and testing. The label will be offset according to the number of time step of forward prediction.

b) Composite model:

22 features mentioned in the table are extracted from all of the stocks in the community 0. Since there are 4 stocks in community 0, the feature matrix will have 22x4, that is 88 features for the time series AAPL.

We build these two models to show that the community-based graph approach has more information and would eventually perform better in the long run. The figure 26 shows the 12 communities derived from the stock graph.

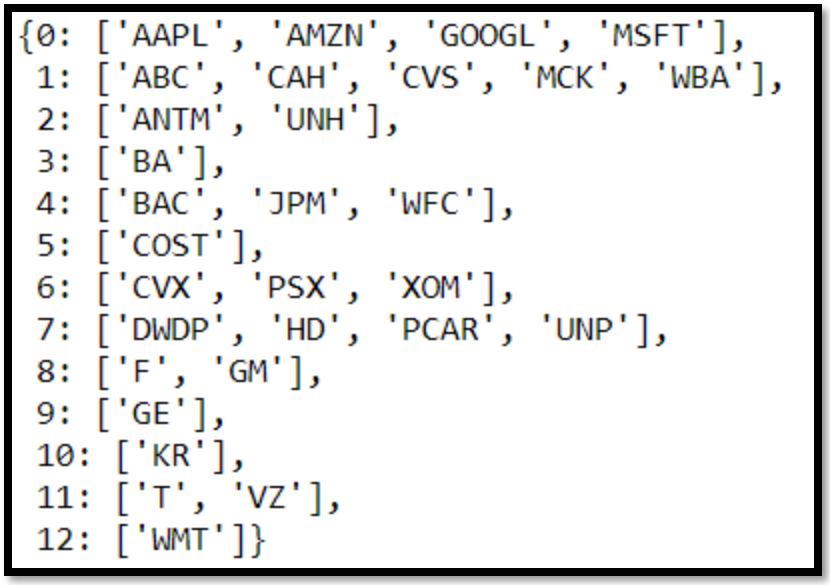


Figure 26: Stocks aggregated based on communities

III. Statistical Model

In this study we have also implemented a traditional statistical model, auto regressive integrated moving average (ARIMA) which is very robust and generic for any type of time series forecasting. ARIMA is an acronym which breaks down as:

AR: *Auto-regression*. This models the relationship between the lagged parameters of the time series as variables.

I: *Integrated*. Differencing is integration is done to remove the stationarity in the time series. A time series is non-stationary if it has unit root. A stationary time series has a mean of 0 and std. deviation of 1.

MA: *Moving average*. This models the residual error terms as lagged parameters from moving average model.

Thus, ARIMA has 3 parameters, p, d and q.

- **p**: Number of lagged terms considered in building model.
- **d**: Number of times the consecutive terms are differenced to make time series stationary.
- **q**: This is the length of the moving average window.

If the value of either of p,d,or q is 0, then that component is not considered in building the model. Therefore, depending on the parameter value, we can build ARMA, AR, MA, I, or ARIMA model to fit the given time series.

Chapter 7: Experiments.

I. Metrics

In this project we have built multiple models for each stock, and there are total 30 stocks in the graph approach. Therefore, in the linear models, 30 models need to be built as well, 30 single models and another 30 for composite models. Evaluating, visualizing and comparing the results of these many models in difficult. Therefore, all the metrics used in this project have been merged to a single value for every time step forward prediction.

Since, all the stock prices are at different scale, the absolute error will be at different scales as well. To solve this problem, all the time series have been normalized, making all the stock prices in the range of 0 and 1. This makes the merged results from different models comparable.

In this project we have used the following metrics:

1) Root mean squared error (RMSE)
2) Mean absolute percentage error (MAPE)
3) Mean absolute error. (MAE)

a) Root mean square error (RMSE)

RMSE of an estimator is the standard deviation of residual after predicting. Residuals in regression are a measure of how far the regression line is from the data points. Residuals can be thought of as errors after prediction. In the formula below, e_t is the vector calculated by differencing the predicted vector, and the actual values vector or true labels. RMSE is generally used with the continuous vectors of data. Since RMSE squares the error before adding, it gives more weightage to the bigger error values. Therefore, RMSE is better when we want to avoid large errors.

Root mean squared error $\quad \mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$

b) Mean absolute percentage error (MAPE)

MAPE is a measure of prediction accuracy fore time series forecasting. Rather than expressing the model results in mean sum error, MAPE represents the evaluation in terms of accuracy. MAPE can be thought as accuracy in classification models. In the formula below, e_t is the vector calculated by differencing the predicted vector, and the actual values vector or true labels. MAPE gives the percentage value.

Mean absolute percentage error $\quad \mathrm{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{e_t}{y_t}\right|$

c) Mean absolute error (MAE)

In statistical machine learning, MAE is the average of all the absolute error values of prediction and actual value. MAE is easy to interpret and understand as its representation is simple as opposed to RMSE. In the formula below, e_t is the vector calculated by differencing the predicted vector, and the actual values vector or true labels. MAE is generally used with the continuous vectors of data.

Mean absolute error $\quad \mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|e_t|$

II. Experiments:

We conducted experiments on the graph based GCN and linear models leveraging communities in graph. We also conducted experiments using traditional statistical model ARIMA. The input data was minute level data of the stock prices.

a) Graph Convolutional Network:

For GCN, the first input is the graph, G, represented as an adjacency matrix, and another input is the temporal time series at every node. Since there are 30 nodes in the graph, 30-time series were input to the model. Every time series had 17204 data points in total. Time series data cannot be split with percentage for training and testing as it is tightly bound to the days/time. Therefore, 34 days worth of data was used for training and 10 days worth of data was used for testing. This is roughly 75% - 25% train-test split. We train the model for 10 epochs with learning rate of 0.001. We use the optimizer function RMSProp for the gradient descent.

We build 4 GCN models, 1 for each of the input graph. There are 3 graphs generated from Pearson correlation, Spearman correlation, and Kendall correlation, each with an absolute threshold of 0.5. The 0.5 threshold is empirically used in other graph research and we have verified and supported this hypothesis by doing community detection using various threshold and studying the community features like sector and industry of every stock. For the graph generated using spearman rank coefficient, the threshold used is 0.4 and for Kendall Tau it is 0.3. This is done because Spearman rank and Kendall Tau model non-linear relationships and therefore the scale of the coefficient reduces as compared to the Pearson correlation, which models linear relationship. Below you are the figures of the graph given as input. Another graph is generated from the new dataset. This graph is derived from causal relationship rather than correlation relationship. Therefore, we believe

this graph holds huge amount of unknown hidden information than the correlation-based graphs. And this hidden information can be interpreted by GCNs. The figures from 27-30 represent the graphs used in the 4 GCN models.

The inputs to the model are 2 csv files. 1 file for weighted adjacency matrix of the graph and other for time series of every node.

The model outputs the merged metrics RMSE, MAE, MAPE by combining the results of all the normalized time series in the graph. For RMSE the values were added, For MAPE the values were averaged, and for MAE the values were added.

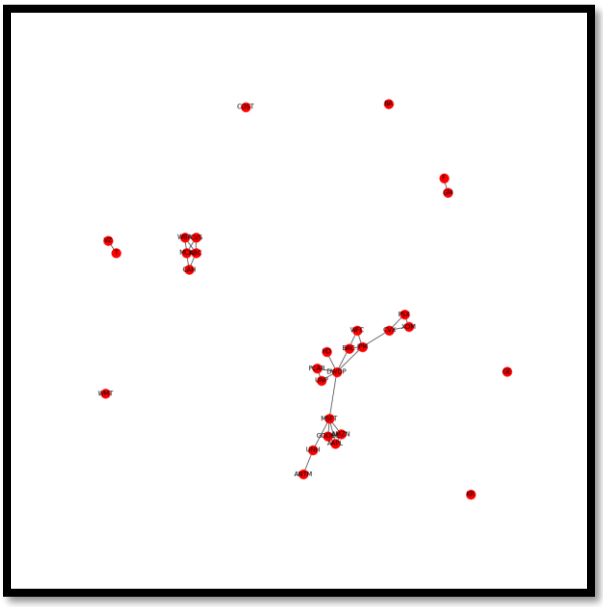


Figure 27: Graph from pearson coef. with threshold 0.5

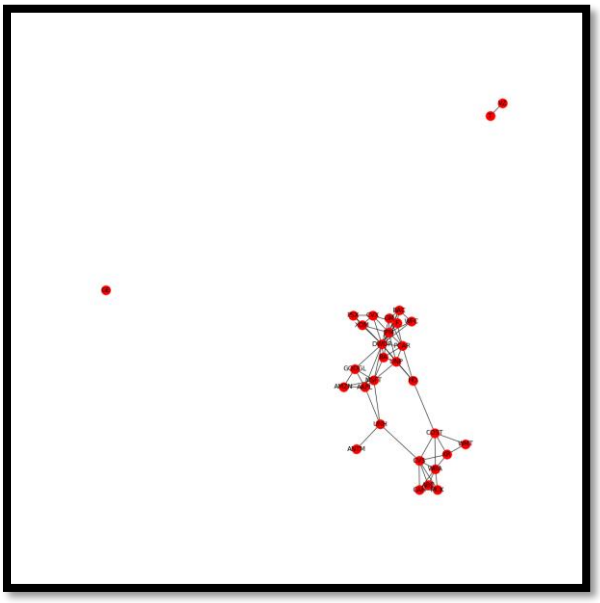
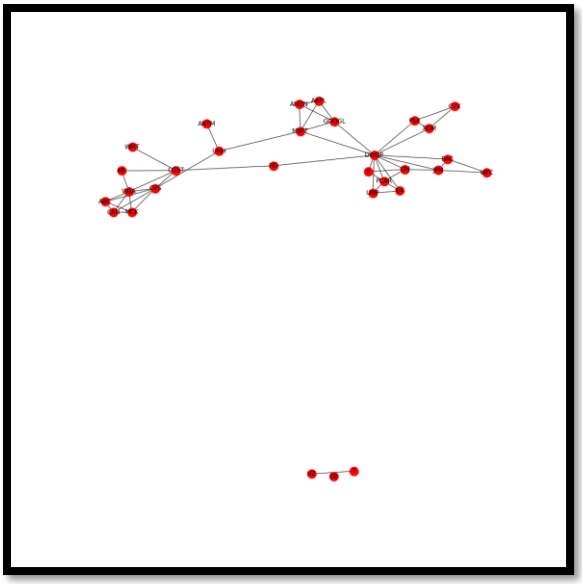


Figure 28: Graph from spearman coef. with threshold 0.4



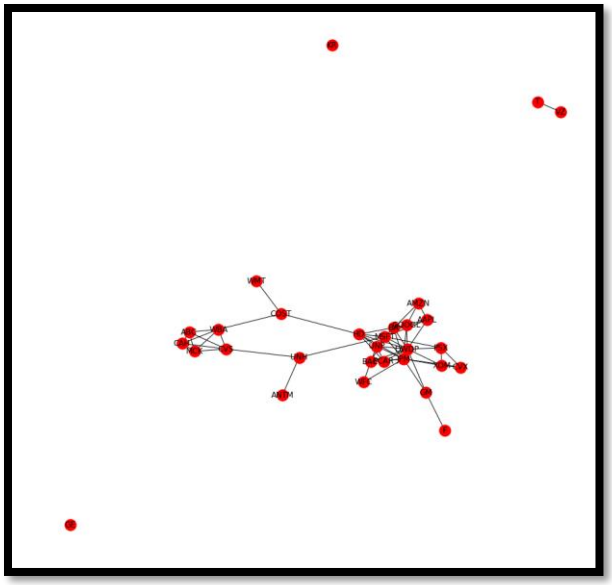


Figure 29: Graph from Kendall tau coef. with threshold 0.3 *Figure 30: Graph from news co-mentions (Causation)*

b) Graph Based Linear Models:

In this experiment, we build linear regression model from sklearn machine learning library. We have two different types of model for every stock:

1) Single model
2) Composite model

Therefore, instead of training 60 different models, we have selected 1 representative stock from each community, and built model for that stock. The representative stock is the one with the highest degree in the graph community. The input time series is normalized to bring all the series to similar scale of 0-1. The input to the single model is a 22-column feature matrix for every data point collected per min for 30 days. The output vector is a 1-dimensional array which stores the forecasted stock price for the nth future time step. Here, n is the offset in the label vector y.

Similarly, for the composite model, we have 22* n number of features in the input matrix where n is the number of stocks in the community. The output vector is a similar 1-d vector with the future predicted values, n time steps ahead.

In the end we merge all the metrics RMSE, MAPE and MAE to get a single representative metric of the linear model.

c) ARIMA

In this part we created an ARIMA model for each of the 30 stocks. The parameter selection is the most difficult and time-consuming part of the model building. To compare multiple ARIMA models with different input parameters (p,d,q), Akaike information criterion (AIC) and Bayesian information criterion (BIC) is used. AIC and BIC explain how well the model with given parameters fits the time series.

In our project we used a grid search for finding the optimal value of p, d, and q parameters. The values of p and q ranged from 0 to 5 and d ranged from 0-2. Therefore, for every stock, 6*3*6 = 108 models were built. The model with the lowest value of AIC and BIC was finalized for that stock. This process was repeated for each of the 30 stocks.

In the end, like other approaches in this project, the results using metrics RMSE, MAPE and MAE were merged to get a single representative value for ARIMA model.

III. Results:

From the above experiments, we build total 7 models for stock price forecasting. Of these, 4 models are GCN models and two models are graph based linear models. The other model is statistical ARIMA model. All the experiments were performed on top 30 stocks from fortune 500. The results for every type of model were merged to get a representative value for that model. This is done for the sake of simplicity in comparing different models. Below are the results for all the models with 3,6, and 9 steps forecasting along with the graphical comparison with the bar graphs.

The best performing model is GCN based on graph constructed from new co-mentions. This is probably because the graph is causation based rather than correlation based. The next GCN model with Kendall tau correlation performs well, probably because of non-linear modelling by Kendall coefficient, the same applies for spearman rank as it models non-linear relationship as well. Clearly, GCN models perform better than the traditional statistical method used for time series forecasting. The composite community based linear model performs better than the single linear model. This supports our initial assumption that graph has rich structural information which can be leveraged not only in time series forecasting but in other problems like classification as well.

Table 6: Results for 3-time steps ahead forecast

Sr. No.	Model	Threshold	Epochs	RMSE	MAPE	MAE
1	GCN with Pearson Correlation	0.5	10	11.919	15.314%	9.096
2	GCN with Spearman Rank Correlation	0.4	10	12.603	5.152%	6.502
3	GCN with Kendall Correlation	0.3	10	10.098	4.054%	5.283
4	GCN based on News Graph (Causation)	-	10	8.673	4.491%	5.245
5	ARIMA	-	-	15.034	21.431%	14.332
6	Single Community based linear model	0.5	-	35.521	3.084%	15.403
7	Composite Community based linear model	0.5	-	32.586	3.387%	15.698

Table 7: Results for 6-time steps ahead forecast

Sr. No.	Model	Threshold	Epochs	RMSE	MAPE	MAE
1	GCN with Pearson Correlation	0.5	10	12.788	15.843%	9.708
2	GCN with Spearman Rank Correlation	0.4	10	13.538	5.623%	7.238
3	GCN with Kendall Correlation	0.3	10	11.564	4.572%	6.219
4	GCN based on News Graph (Causation)	-	10	9.425	5.141%	5.241
5	ARIMA	-	-	17.479	20.667%	11.592
6	Single Community based linear model	0.5	-	44.739	5.612%	22.827
7	Composite Community based linear model	0.5	-	40.866	4.940%	23.018

Table 8: Results for 9-time steps ahead forecast

Sr. No.	Model	Threshold	Epochs	RMSE	MAPE	MAE
1	GCN with Pearson Correlation	0.5	10	13.762	16.135%	10.207
2	GCN with Spearman Rank Correlation	0.4	10	14.602	5.979%	7.912
3	GCN with Kendall Correlation	0.3	10	13.278	5.422%	7.268
4	GCN based on News Graph (Causation)	-	10	11.245	5.113%	6.824
5	ARIMA	-	-	18.756	18.678%	12.435
6	Single Community based linear model	0.5	-	57.617	4.430%	29.581
7	Composite Community based linear model	0.5	-	54.851	3.859%	28.758

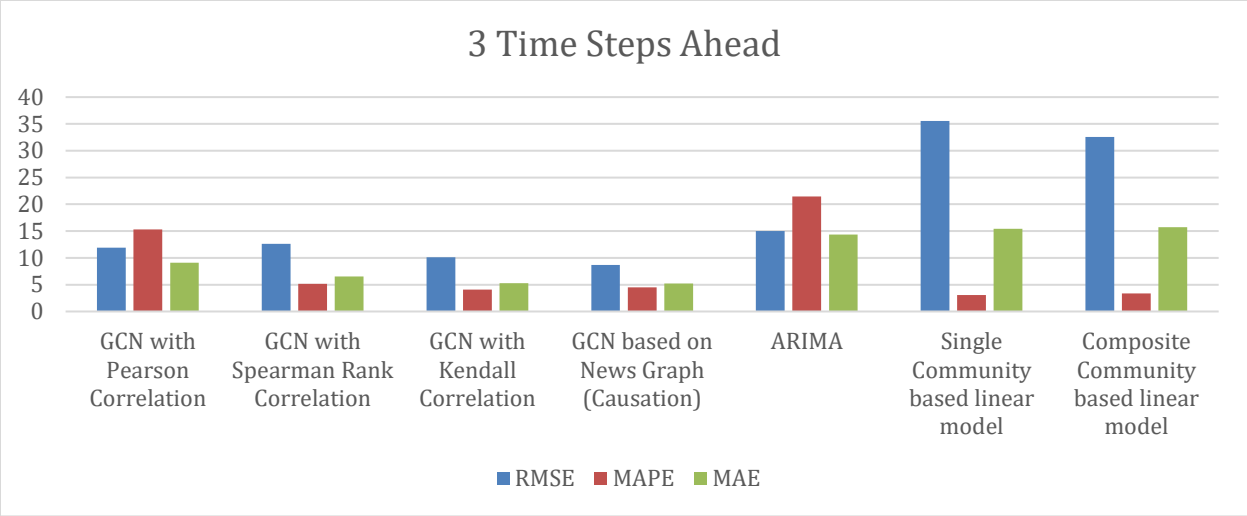


Figure 31: Comparison of all the models using metrics for 3-time step ahead forecasting

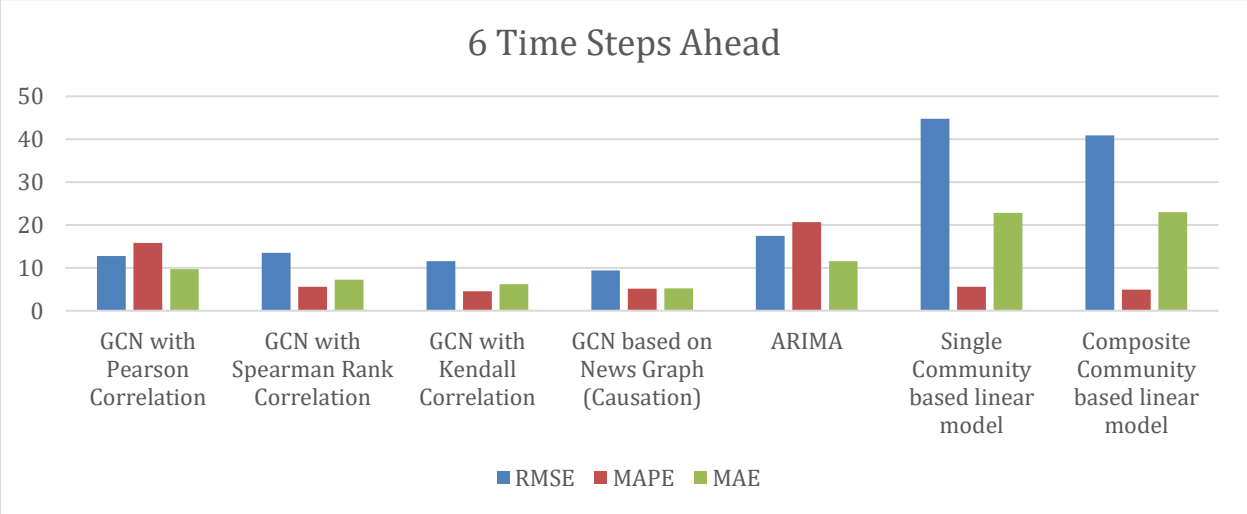


Figure 32: Comparison of all the models using metrics for 6-time step ahead forecasting

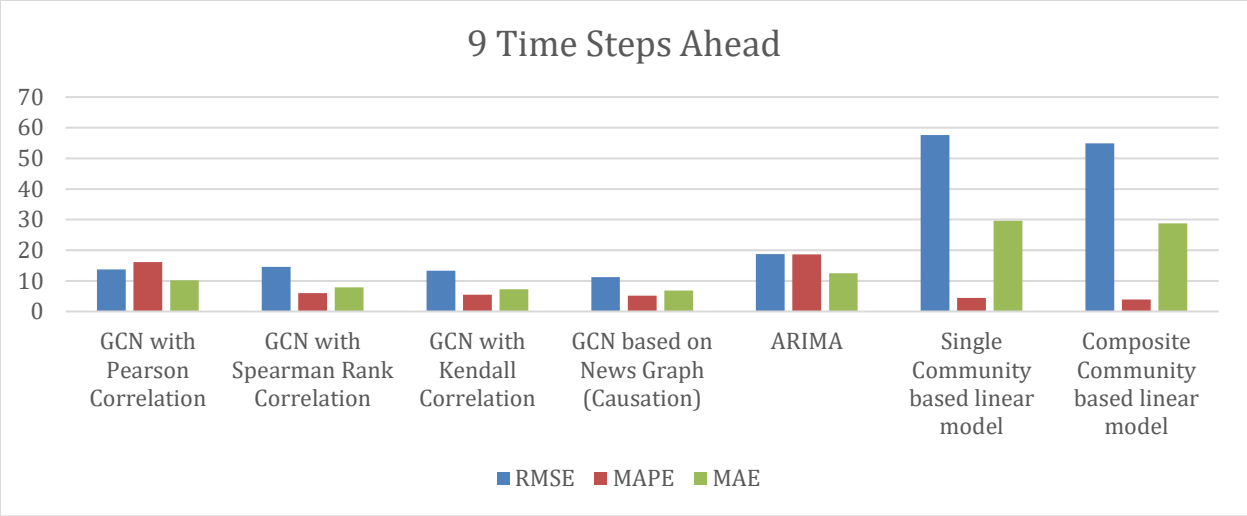


Figure 33: Comparison of all the models using metrics for 9-time step ahead forecasting

Chapter 8: Conclusion and Future Work

In this project we have proposed a novel approach for forecasting stock prices using graphs and leveraging the Spatio-temporal relationship among the companies and its stock prices. In our experiments, overall, the graph-based models perform better than single linear and traditional statistical ARIMA model. We observe that modelling and leveraging graph's spatial structure to temporal features increases the accuracy of forecasting drastically. The experiments prove that graph-based models outperform other traditional and statistical model for stock prediction. In this project we have shown two ways of generating graph, correlation-based and causation-based graphs. We observe that causal relationship holds more hidden information and can be extremely useful. Owing to the resources limitations we used only 30 nodes in the graph, however, the causal based GCN can give extremely accurate forecasts at larger scale with a bigger graph. Through our experiments, we also support our claim that the graph structure holds richer information which gives better results when leveraged. The graph structure can be leveraged in other machine learning problems like supervised and unsupervised classification as well.

Moreover, this Spatio-temporal GCN model for forecasting is not tied to stocks and can be used in any generic time series prediction if underlying graph representation is available.

This project considers 30 nodes graph owing to the complexity of the model and its training period. It will be interesting to study the results and performance of a more complex network (graph). The GCN model proposed in this project is susceptible to exploding gradient problem as nodes with higher degree will have larger value in their convoluted feature representation, whereas nodes with smaller degree will have smaller value in feature representation. A solution to this problem can reduce the complexity of the model training. It will also be interesting to check the performance of GCN on a more generic time series forecasting problems.

References

[1] Rechenthin, Michael David. "Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction." PhD (Doctor of Philosophy) thesis, University of Iowa, 2014.

[2] Milosevic, Nikola. (2016). Equity forecast: Predicting long term stock price movement using machine learning.

[3] http://www.incrediblecharts.com 2019

[4] Singal Vijay, "Beyond the Random Walk: A Guide to Stock Market Anomalies and Low Risk Investing", Oxford university Press, 2003

[5] Morris Stephen and Shin Hyun Song, Oxford Review of Economic Policy, vol. 15, no 3, 1999.

[6] Alice Zheng: Using AI to make prediction on stock market, 2017

[7] Kim Kj. Financial time series forecasting using support vector machines. Neurocomputing. 2003; https://doi.org/10.1016/S0925-2312(03)00372-2

[8]. Huang W, Nakamori Y, Wang SY. Forecasting stock market movement direction with support vector machine. Comput Oper Res. 2005; 32(10):2513-2522. https://doi.org/10.1016/j.cor.2004.03.016

[9]. Ni LP, Ni ZW, Gao YZ. Stock trend prediction based on fractal feature selection and support vector machine. Expert Syst Appl. 2011; 38(5):5569-5576. https://doi.org/10.1016/j.eswa.2010.10.079

[10]. Kumar D, Meghwani SS, Thakur M. Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. J Comput Sci. 2016; 17:1-13. https://doi.org/10.1016/j.jocs.

[11] Qiu M, Song Y, Akagi F. Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. Chaos Solitons Fractals. 2016; 85:1-7. https://doi.org/10.

[12] Preis, T., Moat, H. S., & Stanley, H. E. (2013). Quantifying Trading Behavior in Financial Markets Using Google Trends. *Scientific Reports,3*(1). doi:10.1038/srep01684

[13] R. N. Mantegna, "Hierarchical structure in financial markets," The European Physical Journal B—Condensed Matter and Complex Systems, vol. 11, no. 1, pp. 193–197, 1999. View at Google Scholar · View at Scopus.

[14] H.-J. Kim, I.-M. Kim, Y. Lee, and B. Kahng, "Scale-free network in stock markets," Journal of the Korean Physical Society, vol. 40, no. 6, pp. 1105–1108, 2002. View at Google Scholar · View at Scopus

[15] H.-J. Kim, Y. Lee, B. Kahng, and I.-M. Kim, "Weighted scale-free network in financial correlations," Journal of the Physical Society of Japan, vol. 71, no. 9, pp. 2133–2136, 2002.

[16] J.-P. Onnela, K. Kaski, and J. Kertész, “Clustering and information in correlation based financial networks,” European Physical Journal B, vol. 38, no. 2, pp. 353–362, 2004.

[17] B. M. Tabak, T. R. Serra, and D. O. Cajueiro, “Topological properties of stock market networks: the case of Brazil,” Physica A: Statistical Mechanics and Its Applications, vol. 389, no. 16, pp. 3240–3249, 2010.

[18] M. Tumminello, T. Di Matteo, T. Aste, and R. N. Mantegna, “Correlation based networks of equity returns sampled at different time horizons,” The European Physical Journal B. Condensed Matter and Complex Systems, vol. 55, no. 2, pp. 209–217, 2007.

[19] XuWeichao, YunheHou, Hung Y. S. & Yuexian Zou, 2010. Comparison of Spearman’s rho and Kendall’s tau in normal and contaminated normal models. Manuscript submitted to IEEE Transactions on Information Theory (http://arxiv.org/PS_cache/arxiv/pdf/1011/1011.2009v1.pdf)

[20] Guangxi Cao, Yingying Shi, and Qingchen Li, “Structure Characteristics of the International Stock Market Complex Network in the Perspective of Whole and Part,” Discrete Dynamics in Nature and Society, vol. 2017, Article ID 9731219, 11 pages, 2017. https://doi.org/10.1155/2017/9731219.

[21] M. Gałązka, “Characteristics of the Polish stock market correlations,” International Review of Financial Analysis, vol. 20, no. 1, pp. 1–5, 2011.

[22] C. K. Tse, J. Liu, and F. C. M. Lau, “A network perspective of the stock market,” Journal of Empirical Finance, vol. 17, no. 4, pp. 659–667, 2010.

[23] Souza, TTP & Pappalardo, Giuseppe & Kang, Soong & Aste, Tomaso & Caldarelli, G. Multiplex Structure of Social Media and Financial Networks, 2016

[24] Ghazale, P.P., Zhao, L., Zheng, Q., & Zhang, J. Time Series Trend Detection and Forecasting Using Complex Network Topology Analysis. 2018 International Joint Conference on Neural Networks (IJCNN), 1-7, 2018.

[25] http://francescopochetti.com/wp-content/uploads/2014/10/commu1.png, 2019

[26] https://en.wikipedia.org/wiki/Louvain_Modularity, 2019

[27] Blondel et. al., “Fast unfolding of communities in large networks,” arXiv, 2008. [Available] https://arxiv.org/abs/0803.0476

[28] de Mello Assis, Julia & Pereira, Adriano & Couto e Silva, Rodrigo. Designing Financial Strategies based on Artificial Neural Networks Ensembles for Stock Markets. 1-8. 10.1109/IJCNN.2018.8489688, 2018.

[29] Zhang, Xi & Qu, Siyu & Huang, Jieyun & Fang, Binxing & Yu, Philip. Stock Market Prediction via Multi-Source Multiple Instance Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2869735, 2018.

[30] Jiahong Li, Hui Bu and Junjie Wu, "Sentiment-aware stock market prediction: A deep learning method," 2017 International Conference on Service Systems and Service Management, Dalian, 2017, pp. 1-6 doi: 10.1109/ICSSSM.2017.7996306

[31] Yi Huang, Chien. (2018). Financial Trading as a Game: A Deep Reinforcement Learning Approach, 2018

[32] Yule, G.U. (1926) "Why do we Sometimes get Nonsense-Correlations between Time-Series?" J.Roy.Stat.Soc., 89, 1, pp. 1-63, 1926

[33] https://www.statsdirect.com/help/nonparametric_methods/kendall_correlation.htm, 2019

[34] https://www.semanticscholar.org/paper/Spatio-Temporal-Graph-Convolutional-Networks%3A-A-for-Yu-Yin/4b1c78cde5ada664f689e38217b4190e53d5bee7/figure/0, 2019

[35] https://arxiv.org/pdf/1901.00596.pdf, 2019

[36] https://arxiv.org/pdf/1901.00596.pdf, 2019

[37] https://images.app.goo.gl/fzFbph5Xt9PcSdpx7, 2019