The International Journal of Interdisciplinary Organizational Studies
ISSN: 2324-7649 (Print), ISSN: 2324-7657 (Online)
Volume 19, Issue 1, January-June, 2024

AN INVESTIGATION ON IMPROVED MULTI LEVEL GRAPH ATTENTION BASED GRAPH CONVOLUTION NETWORK FOR STOCK PREDICTION

R Gayathri

Research Scholar, Vellalar College for Women, Erode , Bharathiar University, Coimbatore, Tamil Nadu, India, 2025gayathri@gmail.com

Dr S Devi Suganya

Assistant Professor, Department of Computer Science, Vellalar College for Women, Erode, Tamil Nadu, India, sdevisuganya@gmail.com

Abstract: The issued shares are moved, exchanged, and distributed on the stock market, a capitalistic paradise. Although the issue market serves as the foundation for stock pricing, the stock market's structure and trading activities are far more intricate than the issue market's. As a result, predicting the future accurately becomes a complex and challenging endeavour. Conversely, due to the potential rewards associated with stock prediction, it continues to draw in generations of academics and investors, who in turn continue to develop a wide range of prediction techniques from a variety of viewpoints, theories, investment approaches, and real-world experiences. The majority of current graph-based learning techniques ignore the complexity of stock linkages and instead manually construct stock relationships in order to produce stock graphs. In this study, a sentiment analysis module was created with natural language processes and Gated Recurrent Unit (GRU). The factor analysis module receives the outputs from the LSTM, GRU, BERT, and Relations models from wikidata. The sentiment data on the financial market is included into the factor analysis module. Subsequently, a subset of parameters were fed into the recently created Improved Multilevel Graph Attention network (IMGA) to forecast stock prices. Improved Multi level Graph Attention based Deep Stock prediction network (IMGA-DeSnet) is the name of the suggested technique. Graph Convolution Network (GCN) is used by IMGA to extract relational data. The attention encoded feature representation for certain classes is provided by the factor analysis models' extracted features and trained weights. Finally, the fully connected layer receives the concatenated feature vector and utilises a feature map matrix and input weights to forecast stock. The suggested architecture offers a solid and trustworthy stock prediction method.

Keywords: Stock trend prediction, graph neural network, stock relation, financial news.

I. INTRODUCTION

Nowadays, a lot of people understand the benefits of investing, particularly in light of stock market growth and expansions of online financial services. On the other hand, the stock market is known for its enormous volatility and wealth of data. A lot of people who want to invest financially need to know more about data mining. Because of this, more accurate stock price forecasts help businesses and investors by reducing risk and raising return on investment. Researchers have been interested in stock prediction for a long time. As a valuable indicator of portfolio allocation, accurate stock predictions can assist investors in making wise selections [1]. Since, market participants are assumed to be informationally efficient and transactions occur at fair values, efficient market hypothesis (EMH) holds that stock prices represent markets in full and that stock markets may respond swiftly to information changes [2]. As a result, using the EMH framework to forecast stock prices is impossible. In actuality, nevertheless, this framework's logic is called into doubt. Adaptive market hypothesis (AMH) is then put out by behavioural economists as a revision to the EMH [3]. AMH contemplates that excess returns can result from asymmetrical information implying stocks are predictable, when interactions between individuals and markets are accounted.

In the past, researchers have fitted a linear model to the historical trend in stock prices using statistical techniques. Autoregressive Moving Average Model(ARMA) is set up to evaluate stock prices over time where classical techniques include ARMA, Autoregressive Integrated Moving Average Model (ARIMA), and Generalised Autoregressive Conditional Heteroscedasticity (GARCH). [1]. The ARIMA model, which predicts the overall direction of changes in stock prices, is based on the ARMA model [2]. Wavelet analysis is another feature of the ARIMA model that may be used to match the Shanghai Composite Index [3] more precisely. New stock time series predictions with windows are presented by GARCH models [4]. Concurrently, other researchers combined GARCH and ARMA to create a new forecast which gives volumetric price analysis of multivariate stocks theoretical validity [5]. Conventional methods generally only extract data that is pre-organized. Conversely, traditional prediction techniques need assumptions that are rarely true in real-world scenarios. This makes it challenging to characterise nonlinear financial data statistically.

Many studies use machine learning (ML) techniques like Support Vector Machines (SVM) and Neural Networks (NN) to try and predict stock values which learn from new data and forecast. Many academics utilise the SVM in stock forecasting because it has special advantages when handling small sample sizes, high-dimensional data, and nonlinear circumstances. In terms of stock prediction accuracy, Hossain and Nasser [6] discovered that SVM approaches outperform statistical approaches ones. In order to predict variations in HS300 indices, Chai et al. [7] proposed hybrid SVM models and observed that least square SVMs combined with Genetic Algorithm (GA) performed better. However, when used on voluminous stock data training sets, SVMs consumed a lot of memory limiting its ability to anticipate and resulting financial time series problems were addressed using Artificial Neural Networks (ANN) and multi-layered ANNs. Rapid convergence and excellent accuracy are two advantages of artificial neural networks (ANNs), based on experimental data [8, 9, 10]. Moghaddam and Esfandyari [11] used studies to analyse the influence of numerous feedforward artificial neural networks on market stock price forecasting. Liu and Hou [12] used the Bayesian regularisation technique to enhance the BP (Back Propagation) NN. However, there is need for improvement in the following areas using the standard neural network method: The weak generalisation ability soon causes overfitting and devolves into local optimisation. Improved models are needed to address these issues since a large number of samples must be trained.

Deep learning (DL) offers a new approach to stock price prediction research as artificial intelligence advances. DL models have predicted stock prices and include Convolutional Neural Networks (CNNs) [13], Graph-Based CNNs [14], Recurrent Neural Networks (RNNs) [15], and Long Short-Term Memory (LSTM) [16]. Stock price predictions may be made using RNN mining time series information in data. However, RNN models have gradient dissipations and explosions, which make training complex. LSTM models handle RNN's incapacities in describing time series by implementing time memory functions in cell gate switches [17] and thus perform well in forecasting variations in stock market indices [18]. Yan et al. [19] developed high-precision model based on LSTM deep neural networks for predicting short-term financial markets. Standard RNN and BP neural networks are less accurate than LSTM neural networks in forecasting market prices. The aforementioned models ignored supplementary market information in favour of focusing on one or more pieces of information. Since linked firms' happenings have an impact on stock price swings, stocks do not live in a vacuum.

An attention-enhancing LSTM model combined with data from several sources allowed Zhang et al. [20] for enhanced stock prediction performances. Studies have also exploited graph architectures with the objective of handling data from diverse market sources, though in their early stages. These studies were motivated by financial markets' inherent qualities and correlations between equities. Li et al. [22] used GCN to foresee overnight stock movements with news data to model stock linkages. Chen et al. [21] enhanced prediction accuracies by constructing graphs with important company linkages from company information. Ding et al. [23] used neural tensor networks for extract information from news events and forecasted changes in stock values. Wang et al. [24] improved efficacies of individual stock forecasts and opinion mining. While intricate graphical models have been developed to interpret financial data, existing techniques only incorporate data from many sources into graphs and are unable to effectively use stock market data. Thus, a significant problem remains in effectively mining and integrating multisourced data of stock markets.

Combinations of BERT and Graph Neural Network (GNN) based approaches [25] were studied to anticipate market movements based on news in an attempt to solve the issues listed above. To extract features from stock market data, the three dimensions of text, numerical, and relational data are employed. The BERT model is used to extract textual features of financial markets from news sources. Natural language processing technology that is advanced optimises the components of unstructured news extraction.Multisource data may be efficiently aggregated in the stock prediction challenge utilising graph-based learning approaches. Using the newly developed multilevel attention mechanism, the proposed IMGA can selectively combine representations of nodes obtained from feature extractions for transforming them into stock graph networks, while also assigning different weights to distinct relationships.

II. PROPOSED METHODOLOGY

This section outlines the general structure of our suggested approach before detailing financial market 's text, numerical, and relational data. This proposed system is constructed using corporate relational data in contrast to prior studies that relied on expensive relational modelling techniques. In Figure 1, the general framework is displayed.

Many feature vectors are retrieved using various feature extraction modules following initial data processes and concatenated to construct stock relationship network diagrams using texts and prices. Sentiments on financial markets are then processed using factor analyses. Finally, stock relationship network diagrams are analysed for trend predictions utilising IMGA-DeSnet's multi-layered attention layers. The suggested framework's components are detailed before delving into this revolutionary relational module structures.

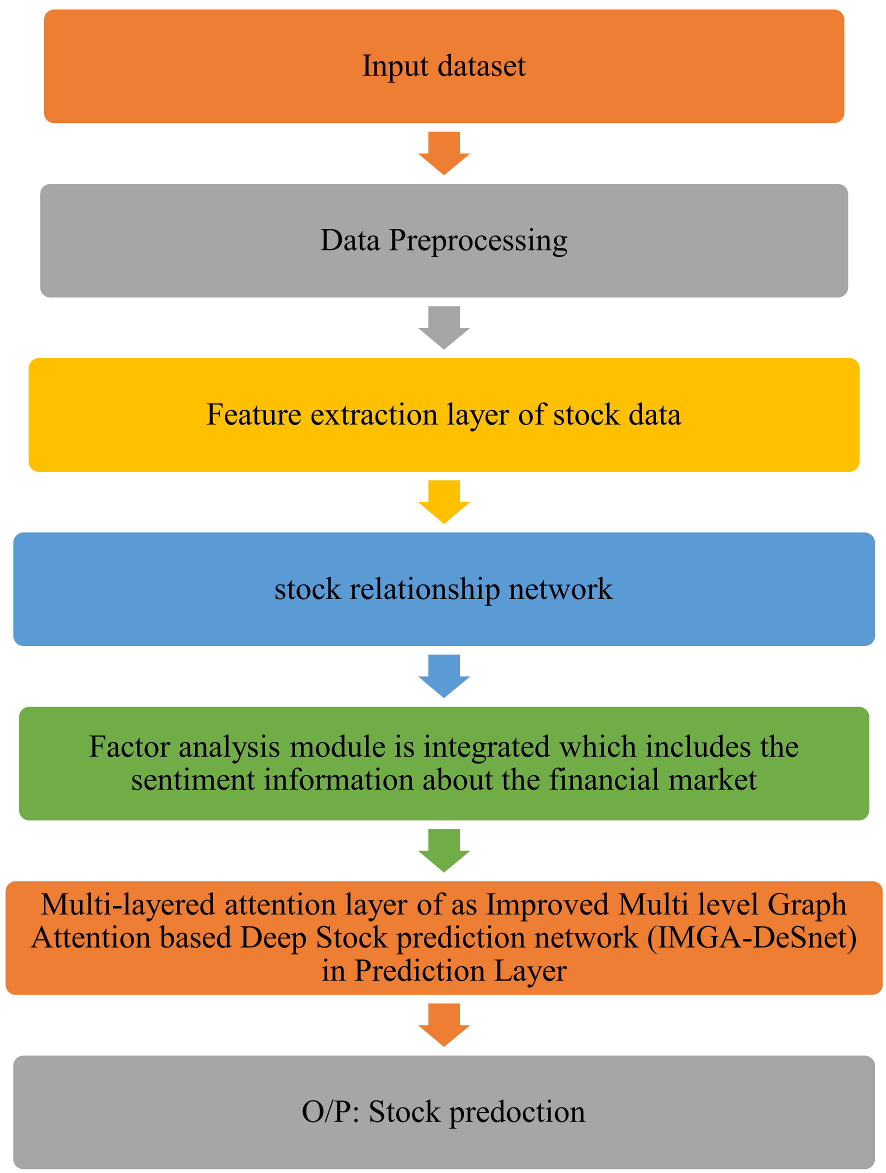

Figure 1. The overall process of the proposed methodology

A. LSTM-BASED ENCODER MODULE

Historical stock prices from financial markets can demonstrate short/long-term reliances. Since, durations and performances of stock prices fluctuate, LSTM-based encoding models which are capable of properly capturing long-term associations are used. Furthermore, in [28], [30], and [40], LSTM were employed for feature extractions.

The feature extraction module takes inputs from a variety of historical stock price raw characteristics, such as starting and closing prices. After preprocessing daily historical closing prices, changes in rates are determined and used as inputs for LSTM in this study. The rate of price changes are calculated. $R_i^t = \{r_i^{t-S+1}, \dots r_i^{t+1}, r_i^t\} \in \mathbb{R}^{S \times D}$ designates historical price changes of stocks i at time steps t. S implies sequence lengths and D signifies feature dimensions at time steps. LSTM network receives time-series data R_i^tas inputs and finally hidden states of outputs, (h_i^t) are utilised as sequential embeds $(e_i^t)(hi^l = e_i^t)$of networks that follow. i.e. the following is obtained

$$i_i^t, f_i^t, o_i^t = f_{\theta^j}, f_{\theta^f}, f_{\theta^e}\left(W[h_i^{-1}; R_i^t] + b_{(i,f,o)}\right)$$

$$C_i^t = i_i^t \odot u_i^t + f_i^t \odot c_i^{t-1}$$

$$u_i^t = \tan\left(W^{(u)} R_i^t + U^{(u)} h_i^{t-1} + b^{(u)}\right)$$

$$h_i^t = o_i^t \odot \tan\left(c_i^t\right)$$

In simple terns, we obtain:

$$E^t = LSTM(\mathcal{R}')$$

Where $\mathcal{R}^t = [\mathcal{R}_1^t, \dots . \mathcal{R}_i^1, \dots . \mathcal{R}_M^t] \in \mathbb{R}^{M \times S \times D}, E^t = [e_1^t, \dots, e_i^t, \dots, e_M^t] \in \mathbb{R}^{M \times U}$represents sequential embeds of M stocks while U stands for embed sizes (LSTM ‘s counts of hidden units).

B. **BERT-BASED ENCODER MODULE**

BERT's input embedding encompasses embedding layers of positions, segments, and tokens. Each word is converted into a set dimension by token embeds. Before each input, a unique token named "[CLS]" appears. The segment embeddings primarily discriminate between two types of sentences, each of which has the input token "[SEP]" assigned to it. Training yields the position embedding.

In order to provide a more detailed example of the BERT extraction of stock news text data features in the financial field, we will first divide the news collected from stock I in time period t into T pieces, which we will then feed into the pretrained encoding model as $N_i^t = \{n_i^{t-T+1}, \dots, n_i^{t+1}, n_i^t\} \in \mathbb{R}^{T \times H}$, where T stands for text paragraph counts and H implies texts feature dimensions. BERT encoders truncate or complete input sentences, preficing [CLS] symbols to sentences and attaching [SEP] symbols at ends. Related news items N_i^t for stocks I are transformed into [CLS] n_i^t+ [SEP], and semantic features of phrases arte represented by output vectors corresponding to [CLS], where semantic feature informations of sentences can be represented by output vectors corresponding to [CLS]. Following varied deeper level feature learning operations of BERT models, [CLS] vectors of previous layer classification output results accurately represent semantic information of texts. For news text features, top layers of output classification need to obtain feature vectors from final transformer layers before phrases are retrieved. i.e. we obtain:

$$F^t = BERT(\mathcal{R}')$$

Where $\mathcal{M}' = [N_1^t, \dots . N_i^1, \dots ., N_M^t] \in \mathbb{R}^{M\times T\times H}, F^t = [f_1^t, \dots, f_i^t, \dots, f_M^t] \in \mathbb{R}^{M\times V}$represents the text feature embedding vector of M stocks, and V represents the embedding size.

C. RELATION EXTRACTION MODULE

Initially, we extract the enterprise relations from the publicly available Wikidata data in the relation extraction module. Wikidata stands for heterogeneous graphs with varying nodes and edges that maintain interactions between diverse entities (such as nations, businesses, and individuals). We are solely interested in the stock-related company node types in our investigation. Nonetheless, these businesses frequently have a variety of edges, and their interactions are quite limited. Inspired by [34], we tackle this challenge by dealing with heterogeneous networks using meta-paths minimizing complications in heterogeneous graphs into homogeneous ones having only nodes pertaining to stocks (businesses).

Wikidata is represented as heterogeneous networks $G = (V, E, T)$, where nodes v and links e are dispersed by mapping functions $\varphi(v) : V \rightarrow T_V$and $\phi(e) : E \rightarrow T_E$ are associated. $T_V \; and \; T_E$ imply collections of nodes and relations. Relationship extraction modules learn d-dimensional latent representations $X \in \mathbb{R}^{|V|\times d}$ $(d \ll |V|)$ acquire structural relationships between targetted nodes.

Subsequently, meta-path schemes leverage on meta-path bootstraps with just two hops to direct random walk processes in heterogeneous networks which are explained: $\mathcal{P}: V_1 \overset{R_1}{\rightarrow} V_2 \overset{R_2}{\rightarrow} V_3$. Probabilities of transitions at steps i can be defined as

$$p\left(v^{i+1}\middle|v_t^i, \mathcal{P}\right) = \begin{cases} \dfrac{1}{\left|N_{t+1}\left(v_t^i\right)\right|} & \left(v^{i+1}, v_t^i\right) \in E, \Phi\left(v^{i+1}\right) = t+1 \\ 0 & \left(v^{i+1}, v_t^i\right) \in E, \Phi\left(v^{i+1}\right) \neq t+1 \\ 0 & \left(v^{i+1}, v_t^i\right) \notin E \end{cases}$$

where$v_t^i \in V_t$ and $N_{t+1}\left(v_t^i\right)$ represent the V_{t+1} type of the field of node v_t^i.

D. RELATION GRAPH MODELLING

In isomorphic graphs modified by relational extraction module in earlier sections, the relational graph module updates nodes, or stocks (businesses). The primary roles of GNNs are information exchange amongst neighbouring nodes, information aggregation amongst neighbouring nodes, and lastly, supplementing each node's representation with information from surrounding nodes. Node attributes are essential to the graph prediction job's execution, thus we need to effectively combine different kinds of relationship data that were collected from various nodes. This paper suggests a novel multilevel graph attention network for relational graph modelling (ML-IMGAT) based on GNNs in order to achieve this. The precise structure of ML-IMGA stock fluctuations is shown in Figure 2.Our model can gather information on different connection types from different nodes and filter out information that is invalid for trend prediction by adding many layers of attention mechanisms at different levels and giving information screening varied weight values. Our approach is essential for effectively forecasting since there are several varieties of stock connections, some of which have nothing to do with market forecasting.

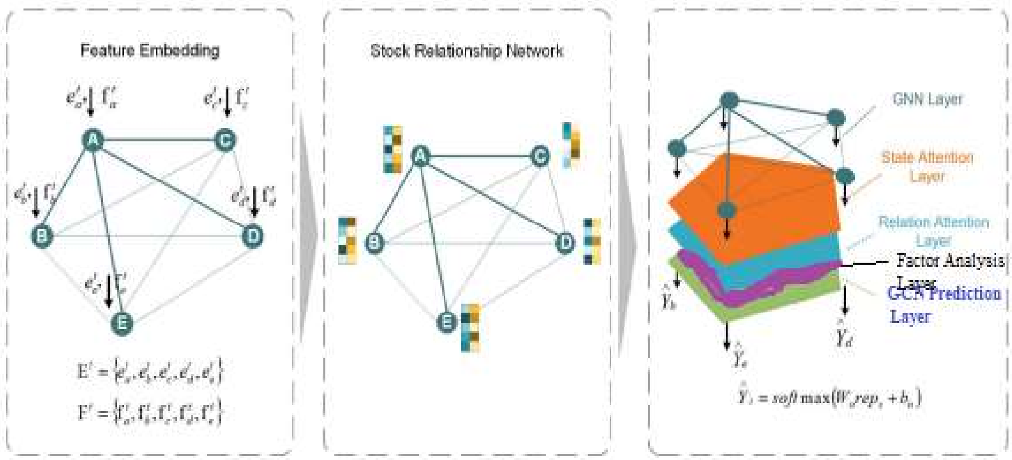

FIGURE 2. Overview of the ML-IGAT mode

Price embed vectors $e_i^t \in \mathbb{R}^h$ and news text embed vectors $f_i^t \in \mathbb{R}^h$ of stocks i at times t generated utilising stock data feature extractions. For convenience, the superscript t is omitted consistently in following texts, assuming all vector representations are generated at the same time t. Furthermore, because implementations are based on GNN, knowledge on collections of targetted nodes' neighbouring relation types I. Embedded vectors of relationship types $r_m \in N_i^{r_m}$and relation types m of nodes i as $r_m$as$e_{rm} \in \mathbb{R}^d$. Our objective is to gather specific relationship-related data about nearby nodes.

ML-GAT selects relevant data with the same sort of connection from a set of nearby nodes inside the first-level state attention layer. The attention mechanism's main job is to calculate different weights according to the selected relationship type, r_m. By joining the node representations of nodes i and j with the relation type embedding vector e_{rm} to create a vector, where $j \in N_i^{r_m}$, is required prior to computing the state attention coefficient. The connection vector as $x_{ij}^{r_m} \in \mathbb{R}^{2h+d}$ is what we refer to. The following formula may be used to calculate the attention coefficient of the relationship type r_m between target node i and target node j:

$$\propto_{ij}^{r_m} = \frac{exp\left(LeakyReLU(\vec{a}_r^T\left(W_s x_{ij}^{r_m} + b_s\right))\right)}{\sum_{k\in N_i^{r_m}} exp\left(LeakyReLU\left(\overrightarrow{a}_r^T\left(W_s x_{ik}^{r_m} + b_s\right)\right)\right)}$$

where $W_s \in \mathbb{R}^{2h+d}$ and $b_s \in R$ are learnable parameters that calculate attention coefficients. Firm connections' output characteristics are generated as weighed averages of all nodes using the algorithm below based on determined state attention coefficients.

$$Z_i^{r_m} = \sigma\left(\sum_{j\in N_i^{r_m}} \left(\propto_{ij}^{r_m} W_s e_j\right)\right)$$

The vector representation of any relationship between stocks is generated by calculating the above formula, and this may be thought of as the relationship's summary information. The relationship type $r_m$of stock i's summary data is contained in the vector $Z_i^{r_m}$. A summary of all the target stock company's subsidiaries, for instance, may be found in the subsidiary relationship vector representation. Similar to how people make investing decisions, our model ought to sort through all of the available data and give priority to the information that is crucial for trading decisions. Hence, in ML-GAT second-level relation attention layers, attention methods are used to assign significant weights to various bits of sign information. Concatenation of summary information vectors $Z_i^{r_m}$ of relationships, current node representations e_i, f_i of stock companies and embed vectors of relationship types e_{rm} and denote concatenated vectors $v_i^{r_m} \in \mathbb{R}^{2h+d}$ as feature inputs for next attention layers. The attention coefficient is determined at the second-level attention layer using the formula, which is similar to the preceding formula

$$\propto_i^{r_m} = \frac{exp\left(LeakyReLU(\vec{a}_r^T\left(W_r v_i^{r_m} + b_r\right))\right)}{\sum_{k\in\varphi} exp\left(LeakyReLU\left(\overrightarrow{a}_r^T\left(W_r v_i^{r_k} + b_r\right)\right)\right)}$$

where$W_r \in \mathbb{R}^{2h+d}$ and $b_r \in R$ are the learnable parameters. An aggregated relation representation is produced by adding the weighted vectors of each kind and using a sigmoid function to get e weighed averages of source-hidden features using:

$$e_i^r = \sigma\left(\sum_{k\in\varphi} \propto_i^{r_k} W_r Z_i^{r_k}\right)$$

Finally, nodes and their node representations are added to obtain representations of targets

$$rep_s = e_i^r + e_i + f_i$$

E. FACTOR ANALYSIS MODULE

Wikidata's corporate relationship data, financial markets, and investor sentiment were integrated into this phase by examining the textual data from stock forum comments. Prediction accuracy will increase with the addition of sentiment indicators to the model's characteristics. But using a broad lexicon in the finance industry isn't going to provide positive outcomes. As a result, creating an emotion dictionary unique to each stock is required. Based on reputable financial investment dictionaries, a sentiment dictionary is created that is appropriate for sentiment indicator computation and individual stock sentiment analysis. Beyond classifying emotions as positive or negative in sentiment analysis, additional factors including macroeconomic circumstances and changes in policy will also affect the prediction of stock price.In this study phase, emotional signs including grief, fear, rage, and contempt were also taken into consideration.

Once information from various data sources has been acquired, effective factors of financial indicators, technical indicators, and public opinion indicators are considered while designing the factor store. This study uses the Markov method to determine the optimal feature combination from financial indicators, technical indicators, relational variables, and sentiment components based on prior research experience. Markov processes are stochastic processes that meet Markov properties. Stated differently, it is a process for which future occurrences can only be predicted from its current state, and these predictions are equally accurate as those derived from an understanding of the process's whole history. Another popular way to characterise a Markov chain is as having discrete time in a countable or continuous state space (i.e., independent of the state space). Markov chain as a countable state space (i.e., time independent) Markov process in discrete or continuous time. One of the extra layers of the suggested deep learner model is the factor analysis module, which uses the Markov process.

The most basic Markov model is the Markov chain. It depicts a system's condition across either continuous or discrete time. The observer may fully witness every state of the model. A family of random variables, $\{X_n\}$ where n = 1, 2, 3.... is referred to as a stochastic process. The value x_n of random variables $X_n$at times n called states of random variables at that time.State spaces S are collections of all potential values that random variables X_n may take.

When the dependence of x_{n+1} is entirely captured by the dependence on the last sample x_n we say that the stochastic process is a Markov chain. More precisely, we have:

$$P(X_{n+1} = x_{n+1}|X_1 = x_1, \ldots, X_n = x_n) = P(X_{n+1} = x_{n+1}|X_n = x_n)$$

where: $x_{n+1}, x_n, \ldots, x_1 \in S$

Which is called Markov Property.

We can automatically determine the transition probabilities when they are unknown by looking through the system's previous data. The approach used in this thesis looks for a collection of pairings inside a set of state sequences. The final model's transitions are represented by pairs, which may be expressed as (s_i , s_j).. In order to get pi,j, we must divide the total number of pairings with i as the beginning point by the number of times we see a transition from state i to j. Let's provide a formal definition for the previously discussed concepts:

$$s_0, s_1, \dots, s_n \in S$$

$$Q = (q_0, q_1, q_2, \dots . q_T) | q_i \in S$$

Where S stands for sets of states and Q are sequences of states observed in single executions of systems. Collections of various sequences with varying lengths can be acquired by monitoring systems K times. $T_0, T_1, \dots, T_K$ and on collecting all these sequences in D, sets of all data result in:

$$Q_k = \left(q_0^k, q_1^k, q_2^k, \dots, q_{T_k}^k\right)$$

$$D = \{Q_0, Q_1, Q_2, \dots, Q_K\}$$

The next step explodes sequences as sets of pairs. Pairs (s_i, s_j) represent associations between $s_i \; and \; s_j$ based on obtained sequences and generated pairs are added to set F.

$$C_k = \left\{\left(q_t^k, q_{t+1}^k\right) | \forall t \in 0{:}\, T_k \Lambda \forall k \in 0{:}\, K\right\}$$

$$= \left\{(s_i, s)j)\middle| \forall t \in 0{:}\, T_k \Lambda \forall k \in 0{:}\, K, q_t^k = s_i \Lambda q_{t+1}^k = s_j\right\}$$

$$F = C_0 \cup C_1 \cup \dots \cup C_k$$

In the end, the probability to reach the state j by being in the state i can be computed as follows:

$$p_{s_i s_j} = p_{i,j} = \frac{\left\|\{(s_i, s_j)\middle|(s_i, s_j) \in F\}\right\|}{\|\{(s_i, x)|(s_i, x) \in C \Lambda x \in S\}\|}$$

The number of marriages with both s_i and s_j is counted in the numerator, while the number of couples with si as the initial member is counted in the denominator.

As a result, the newly created IMGA received the chosen elements for stock price prediction. GCN is used by IMGA to extract relational data. The attention encoded feature representation for certain classes is provided by the factor analysis models' extracted features and trained weights. Finally, the fully connected layer receives the concatenated feature vector and utilises a feature map matrix and input weights to forecast stock.

F. STOCK PREDCITION USING IMGA

Here, GCN (GCN) is used by the IMGA. Convolutional neural networks may be extended to provide graph encoding using GCN. It changes node representations and performs convolution filtering on the network by propagating information between nodes. We stack all three types of constructed nodes within node set V. On a graph, the GCN layer may be written as a non-linear function $f(\boldsymbol{V}, \boldsymbol{A})$, where the set of nodes V are represented as d dimensional vectors $\mathbf{V} \in \mathbb{R}^{(N+R+T) \times d}$.. When many GCN layers are stacked and the convolutional approach provided by Kipf and Welling (2017) is used, the GCN may be represented as follows:

$$V^{l+1} = f(V^l, A) = \delta(AV^l W^l)$$

where δ(•) is an activation function; in our investigations, LeakyReLU is the function of choice. Generally speaking, the layer number is indicated by the superscript l. The convolutional filter's learnable parameters are denoted by $\mathbf{W}^l \in \mathbb{R}^{d \times d}$. Superscript l is used to identify the convolutional filters of various graph convolution layers. The adjacency matrix may be multiplied to add the characteristic of each node's neighbour, as can be easily observed. For example, Equation (2) might be expressed differently if entity node 5 includes two neighbouring nodes: sentence node 1 and mention node 4.

$$V_5^{l+1} = \delta\left(a_{35} v_3^l W^l + a_{45} v_4^l W^l + a_{55} v_5^l W^l\right)$$

where V_i^l1 indicates the node representation corresponding to the i-th node of the l-th layer, and a_{ij}represents the element at row i and column j of the neighbouring matrix A. In this instance, the updated node feature is obtained by first passing the result through one activation function, which is followed by the GCN aggregating all neighbouring node information with the same convolution weights. In this manner, neighbouring nodes influence one another in the network, and following several layers of convolution operations, the relationship between entity nodes is discovered.

where indicates the element at row i and column j of the neighbouring matrix A, and the node representation corresponding to the i-th node of the l-th layer. In this scenario, the updated node feature is created by first passing the result through one activation function, and then using the same convolution weights in the GCN to aggregate all neighbouring node information. Neighbouring nodes influence each other in this way inside the network, and after several layers of convolution operations, the link between entity nodes is found.Factual evidence supports the notion that graph convolution is a type of Laplacian smoothing that combines the properties of the node and its neighbours (Kipf and Welling, 2017; Li et al., 2018).The smoothing process makes the features of nodes in the same cluster similar, which optimises the classification task, the reason for lower performances of GCNs. Nevertheless, stacking many GCN layers may result in the issue of over-smoothing. Oversmoothing issues may result in node representations that are too similar, eliminating the node's ability to be distinguished in the classification function. Furthermore, this issue restricts the model's capacity to simulate long-distance relations. Nonetheless, the document network we create frequently involves long-distance reasoning routes as it may be necessary to deduce the relationship between entity nodes from a number of mention and sentence nodes.

To mitigate the aforementioned issues, we suggest a gating method specifically designed for GCNs. The conventional graph convolutional layer is split into two phases by this approach. In line with how conventional GCNs operate, the initial phase involves aggregating the structural information of neighbouring nodes in order to pass messages on the graph. Using a gating mechanism to regulate the updating of node representations is the second stage. Here is a formula for calculating the gating mechanism:

$$g_l = sigmoid(W_g V_{l+1} + U_g V_l + b_g)$$

$$V_{l+1} = V_{l+1} \odot g_l + V_l \odot (1 - g_l)$$

where $\boldsymbol{W}_g \in R^{d\times d}$ and $\boldsymbol{U}_g \in R^{d\times d}$ are two learnable parameters. The gate $\boldsymbol{g}_l$ controls the new node representation $\boldsymbol{V}_{l+1}$ update of each layer by considering the node representations generated by the previous layer $\boldsymbol{V}_l$ and the current graph convolutional layer $\boldsymbol{V}_{l+1}$.

The gating technique aims to store the distinct local information associated with the current node representations following each graph convolution operation. The model learns more distinctive node representations and enhances its comprehension of the document graph when paired with distinct local information and efficient global information, which lessens the over-smoothing problem caused by multi-layer GCNs. Moreover, our proposed gating strategy does not require human hyperparameter modification to determine each hop's contribution, in contrast to the walk-base method's edge update mechanism (Christopoulou et al., 2019b).

For the target stock price prediction job, we deploy a shallow neural network implementation after many instances of aggregation. The prediction job is represented as a classification problem: we split the future price trend into three categories: up, neutral, and down, and the task parameters are given in detail subsequently. The prediction network is a basic linear transformation layer with the following definition:

$$P(r|e_i, e_j) = softmax(e_i^T W_{cls} e_j)$$

where $\boldsymbol{W}_{cls} \in R^{d\times k\times d}$ are learnable parameter matrices. *k* represents target prediction counts, which is 2 and in this work causes relation extractions are binary classification problems.

Stochastic gradient descent (SGD) algorithm is used in this work to minimize log likelihood functions where loss functions are:

$$\mathcal{L} = -\sum_{d\in\mathcal{T}} \log p(r_{e_i,e_j=r^*_{e_i,e_j}} |d$$

where $\mathcal{T}$ represents the training set of full stock dataset, $r^*_{e_i,e_j}$is the gold label for the relation between entity on the full stock relationship (e_i, e_j) in document *d*. During training, we minimize the loss function L of the stock prediction.

III. EXPERIMENTAL RESULTS

In this part, we carry out in-depth studies to examine the merits and drawbacks of the suggested approach and contrast it with a number of well-liked, recently developed financial strategies. Prior to reporting the outcomes of each experiment, we first describe the experimental data sources and experimental parameters.

A. DATA GATHERING AND PREPARATION

Details of prices of established stock markets around the globe that are the subject of our research comprise 423 stocks from the S&P 500 index and 286 stocks from the CSI 300 index, respectively. We utilise the China Stock Market & Accounting Research Database for equities in the CSI 300 index and the Yahoo Finance website (https://finance.yahoo.com/) for companies in the S&P-500 Composite index. Our relational data, which comes from Wikidata and does not include every stock in the index, excludes certain stocks from Wikidata that are unrelated to other companies. For the S&P 500 index, for instance, we removed the irrelevant stocks and added the stocks of the 423 remaining firms as target stocks..

TABLE 1. Statistics of historical price data

Index	Stocks	Training Days	Validation Days	Testing Days
S&P 500	423	08/02/2013-23/05/2017 1080days	24/05/2017-27/03/2018 213days	27/03/2018-29/08/2019 316days
CSI 300	286	08/02/2013-23/05/2017 1080days	24/05/2017-27/03/2018 213days	27/03/2018-29/08/2019 316days

Many studies get original historical price characteristics, such as opening and closing prices, to dedicated feature extraction modules, and these historical price changes are used as input to LSTM. The rates of changes in price of stocks at times t may be calculated as: $R_i^t = \frac{P_i^t - P_i^{t-1}}{P_i^{t-1}}$where P_i^t and P_i^{t-1} are the closing prices of stock i at time t and t-1, respectively.

Financial news data:

For financial news, we extract news related to the target stock from websites such as Yahoo Finance within a specific time interval, and our final text dataset consists of nearly 150,000 texts about the target stock.

Corporate relation data:

Wikidata are key sources of entity relationships; they have large corporate entiy counts and commercial links that may effect stock prices. Third types of data are relational data from Wikidata (https://www.wikidata.org/wiki/Wikidata: Main_Page). Currently 48 million items (Google Inc.) and hundreds of millions of words (Apple, which was founded by Steve Jobs) in Wikidata, free collaborative knowledge bases. Motivated by [28], meta-paths with a max. of two hops extract 62 types of second-order relations and 9 types of first-order relations from [35]..

B. EXPERIMENT SETTINGS

The price change rate based stock price training data is divided into three categories namely up, neutral, and down. Calculations of stock price change rates have also been explained before.The price change labels for training sets have been modified as follows:

$$f = \begin{cases} up & R_i^t \geq r_{up} \\ neutral & r_{down} \leq R_i^t \geq r_{up} \\ down & R_i^t \leq r_{down} \end{cases}$$

Here, we set $r_{up} = 0.6, r_{down} = -0.6$.

Numerous research on trading tactics have been undertaken [41], [42]. In order to calculate the profitability of the methodology given in this paper, we simulate stock trading using a few prominent trading approaches. We created a portfolio based on the predicted values from the prediction model. Because the model is divided into three classes during training, the prediction vector is three-dimensional, with each dimension reflecting the expected probability of a class. Every day, the stock is purchased, sold, or held. If there is a high likelihood of an upward increase, the stock is acquired at the day's closing price with all available funds. If the likelihood of a decrease is greater, the stock is sold at the transaction's closing price. Otherwise, no transaction occurs on that particular day. Then, using TensorFlow, we design our model, and then use the Adam optimizer to modify parameters across 100 epochs on a single NVIDIA Tesla K80 GPU [37]. The learning rate was set to 5e-4, the weight decay to 5e-5, and the batch size to 32. To minimise overfitting, we employ dropout [36] at the end of each layer and set its value to 0.5. Our activation function is a leaky ReLU. Because the results of each iteration of the experiment vary substantially depending on the actual conditions, we ran each experiment 10 times separately.

C. EVALUATION METRICS

Typical criteria were chosen to evaluate model categorizations and profitabilities to compare performances of proposed techniques to benchmark models. Stock trend predictions are common predictions and initially two assessment indicators that are commonly used in classification tasks namely : accuracy and F1-score were selected in this work and computed using the formulas

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision}$$

Macro F1 values are obtained by averaging computed F1 scores of categories. To evaluate profitabilities of proposed methods, the following two metrics were used in comparisons of methods for profitabilities. The formula used for computing returns of portfolios was:

$$\text{Return}_i^t = \frac{1}{|F^{t-1}|} \sum_{i \in F^{t-1}} \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$$

whereF^{t-1} denotes groups of stocks included in portfolios at times $t-1, p_i^t$denotes prices of stocks i at times t, $|\cdot|$ denotes item counts in set portfolios. Sharpe changes in rates are indicators that consider both returns and risks which help in measuring performances of investment risks compared to returns and computed as:

$$\text{Sharpe}_a = \frac{E\left[R_a - R_f\right]}{\sigma_p}$$

where R_a denotes portfolio rates of returns, R_f denotes risk-free rates, and σ_p denotes standard deviations of portfolio rates of returns where this work used 13-week Treasury bills as risk-free rates.

D. BASELINE METHODS

In this section, we describe several benchmark models chosen for this paper.

(1)MLP One of the most extensively utilised NNs for stock forecasting is the multilayer perceptron. We utilise a basic multilayer perceptron model with four layers in this study, comprising two hidden levels and one prediction layer.

(2)CNN [3] CNNs are commonly utilised because they are quick at modelling time series. A CNN network with three convolutional layers was employed.

(3)LSTM [31] Many prior research have validated the usefulness of LSTM as a frequently utilised DL model for time series forecasting applications. In our studies, we encode the final predictions on historical price data using a two-layer classical LSTM model.

(4)GCN [15] The stock graph is reconstructed using historical information reflecting the target company's connection utilising historical price data as input to the node using a GCN model with two convolutional layers and one prediction layer.

(5)TGC [28] Feng et al.'s temporal graph convolution module is used to represent stock relations. This module encodes the current state of the stock using an LSTM and provides the newest state to the GCN to investigate the linkages between firms. The TGC model summarises all connected information of the target stock's surrounding nodes, whereas the ML-GAT summarises information of different connection kinds and gives varying weights to that information.

Table 2 presents the parameters adopted by the ML-GAT-based method and other benchmark models.

Methods	Parameters and Description	Methods	Parameters and Description
MLP	Hidden layers: 128,64; Optimizer: Adam Learning rate: 0.0001 Epochs: 100	TGC	Hidden: LSTM layers: 64,64 MLP layer: 1 Optimizer: Adam Learning rate: 0.001 Epochs: 100
CNN	Convolutional layer1: 16filters 2*2 Convolutional layer2: 32filters 2*2 Convolutional layer3: 64filters 2*2 Max pooling layer: 2*2 Fully connected layer: 500 Optimizer: rmsprop Learning rate: 0.01 Epochs: 100	GCN	Convolutional layer1: 64filters 2*2 Convolutional layer2: 64filters 2*2 Optimizer: Adam Learning rate: 0.01 Epochs: 100
LSTM	LSTM layers: 60,50,50,50 Dropout layer: 0.2 Optimizer: rmsprop Learning rate: 0.01 Epochs: 100	ML-GAT	The length of time series: 50 Hidden: MLP layer: 2 LSTM layer: 128 Optimizer: Adam Learning rate: 5e-4 Dropout layer: 0.5 Activation function: Leaky_ReLU BERT: BATCH_SIZE=32 MAX_token_LENGTH=128 Epochs: 100

As the GCN's adjacency matrix, we enter each relation type independently. Finally, tables 3&4 give the ten best and ten worst relation types, together with their F1 scores on the test set.

TABLE 3. Results of using the best 10 relations

Relation Type	F1-score
Industry-Legal form	0.4576
Parent organization-Owner of	0.4561
Industry-Product or material produced	0.4552
Owned by-Subsidiary	0.4547
Founded by-Founded by	0.4543
Follows	0.4535
Parent organization	04521
Complies with-Complies with	0.4502
Subsidiary-Owner of	0.4491
Owner of-Parent organization	0.4484

TABLE 4. Results of using the worst 10 relations

Relation Type	F1-score
Board member	0.3112
Lnstance of-Legal form	0.3084
Location of formation-Country	0.3075
Stock Exchange	0.3053
Country-Location of formation	0.2952
Country of origin-Country	0.2948
Country-Board member	0.2886
Country-Country of origin	0.2851
Instance of-Instance of	0.2748
Stock Exchange-Stock Exchange	0.2665

E. WIKIDATA COMPANY-BASED RELATIONS

As shown in tables 5 and 6, we find 9 types of first-order (A R→ B) and 62 types of second-order (A → R1 B ← R2 C) relationships between firms corresponding to the selected stocks in the S&P500. Wikidata entities are represented by A, B, and C, whereas Wikidata relations are represented by R, R1, and R2. The obtained relations are detailed in tables 7&8.

F. Comparison of the Classification Accuracy

TABLE 5. Classification scores for stock prediction tasks

F1-Score							
	MLP	CNN	LSTM	GCN	TGC	ML-GAT	Proposed IMGA

Index	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300
1	0.3462	0.4330	0.3986	0.3329	0.3873	0.4202	0.3989	0.4250	0.4337	0.4449	0.4983	0.5847	0.5211	0.6011
2	0.3331	0.3464	0.3977	0.3013	0.3821	0.4736	0.4307	0.4816	0.4528	0.4268	0.5224	0.66665	0.5331	0.6915
3	0.3728	0.4422	0.3861	0.4549	0.4155	0.5006	0.4142	0.4590	0.4476	0.5431	0.5035	0.5979	0.5231	0.6124
4	0.3661	0.4244	0.4058	0.3654	0.3958	0.4776	0.4192	0.3977	0.44352	0.4809	0.5183	0.6017	0.5301	0.6201
5	0.3749	0.3977	0.3895	0.4514	0.4129	0.5056	0.4335	0.4412	0.4449	0.4841	0.4980	0.5120	0.5121	0.5231
6	0.3776	0.3415	0.3984	0.3316	0.3862	0.3523	0.4011	0.3612	0.4311	0.3607	0.5221	0.5386	0.5341	0.5458`
7	0.3302	0.2758	0.3892	0.3400	0.4104	0.4077	0.4308	0.4811	0.4574	0.3750	0.5078	0.5959	0.5121	0.6121
8	0.3494	0.2770	0.3733	0.4632	0.4138	0.4598	0.4317	0.5305	0.4318	0.4473	0.5386	0.5541	0.5478	0.5701
9	0.3653	0.4279	0.4012	0.3959	0.4120	0.3219	0.3953	0.4184	0.4289	0.5190	0.4941	0.5112	0.5111	0.53112
10	0.3525	0.4244	0.3991	0.3952	0.3882	0.3456	0.3903	0.3206	0.4484	0.4270	0.4985	0.5285	0.5087	0.5367
Average	0.3568	0.3790	0.3939	0.3832	0.4004	0.4265	0.4146	0.4317	0.4412	0.4509	0.5102	0.5691	0.5247	0.5791

Classification accuracy results of several stock indices based on various techniques in stock trend prediction studies are summarised in Table 5 and Figure 3. The results table shows that LSTM outperforms the other two benchmark models in terms of accuracy and F1-score among the three models that do not take into account simulating the connections between stocks. Consequently, we just need to compare the outcomes of our model with those of the LSTM model in order to compare it to the relational modelling module. Every model using relational modelling modules outperforms LSTM in terms of the F1-score. Nevertheless, not all models that take into account simulating the links between stocks outperform LSTM in terms of accuracy, with GCN marginally underperforming LSTM. Specifically, we may conclude that, in ten repeated tests, the proposed IMGA model performs better than the other models on average.

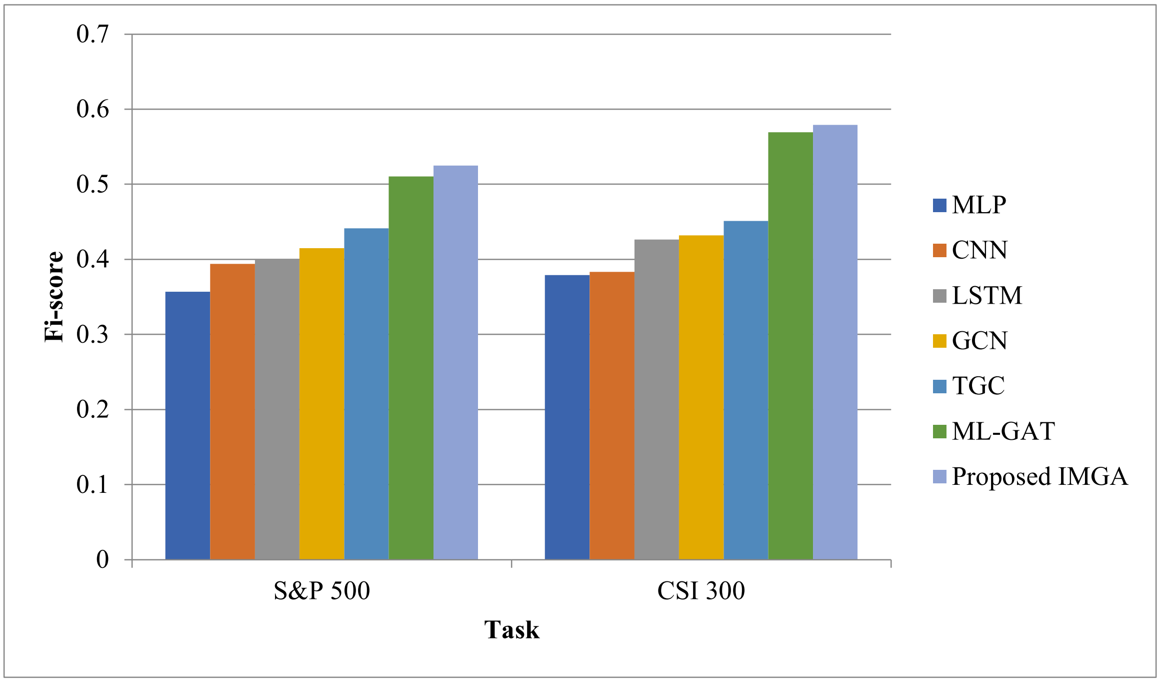

Figure 3: Classification scores for stock prediction tasks

2) PROFITABILITY

TABLE 6. Classification accuracy scores for stock prediction tasks

Accuracy														
	MLP		CNN		LSTM		GCN		TGC		ML-GAT		Proposed IMGA	
Index	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300	S&P 500	CSI 300
1	0.3267	0.3696	0.2912	0.3483	0.4467	0.4335	0.4345	0.3632	0.4779	0.4366	0.5172	0.5264	0.6172	0.6264
2	0.2711	0.3122	0.3423	0.4287	0.4365	0.4123	0.4515	0.4459	0.4674	0.5150	0.4972	0.3985	0.5972	0.4981
3	0.2759	0.2798	0.3601	0.3677	0.4212	0.3270	0.4519	0.4032	0.4443	0.4440	0.5094	0.5126	0.6094	0.46126
4	0.3017	0.4202	0.3497	0.3906	0.4498	0.3912	0.4564	0.4423	0.4498	0.4636	0.5036	0.5866	0.6036	0.6866

5	0.3181	0.4023	0.3136	0.4503	0.4317	0.3650	0.4481	0.4505	0.4569	0.3827	0.5184	0.6507	0.6184	0.7507
6	0.2875	0.3079	0.3543	0.4253	0.4616	0.3931	0.4384	0.4531	0.4473	0.49556	0.5069	0.6162	0.6069	0.7162
7	0.3095	0.4005	0.2885	0.4243	0.4412	0.3751	0.4356	0.2938	0.4694	0.5016	0.5177	0.4912	0.5177	0.5912
8	0.3073	0.3829	0.3429	0.4336	0.4345	0.4444	0.4332	0.4140	0.4665	0.3706	0.5193	0..6524	0.6193	0.7524
9	0.3039	0.3759	0.2827	0.2865	0.4574	0.3980	0.4238	0.3896	0.4769	0.4047	0.4983	0.6249	0.5983	0.7249
10	0.3199	0.3376	0.3528	0.3693	0.4474	0.4882	0.4480	0.4828	0.4693	0.4235	0.4977	0.6383	0.5977	0.7383
Average	0.3022	0.3589	0.3278	0.3924	0.4428	0.4028	0.4421	0.4138	0.4626	0.4438	0.5085	0.5698	0.6085	0.7691

Table 6 presents the profitability results of several models. We compute the portfolio's daily returns using the trading strategies outlined in the prior section. While TGC and the suggested IMGA generated slightly high average daily returns, ML-GAT generated the most competitive daily returns, exceeding TGC and other benchmark models substantially. The most notable finding in this case is that GCN outperforms the model without a relational modelling module in terms of F1-score; however, its Sharpe rate is significantly lower than that of LSTM and other benchmark models with unrelated modelling modules, whereas TGC's Sharpe rate is higher than that of all other benchmark models outside of the Proposed IMGA. Based on the variation of the results of multiple tests, the Sharpe rate variance of several sets of benchmark models is much larger than our model. In the profitability test, this demonstrates that our model has more reliable qualities. To summarise, the proposed IMGA model yields notable results for both the Sharpe rate and the expected average daily returns.

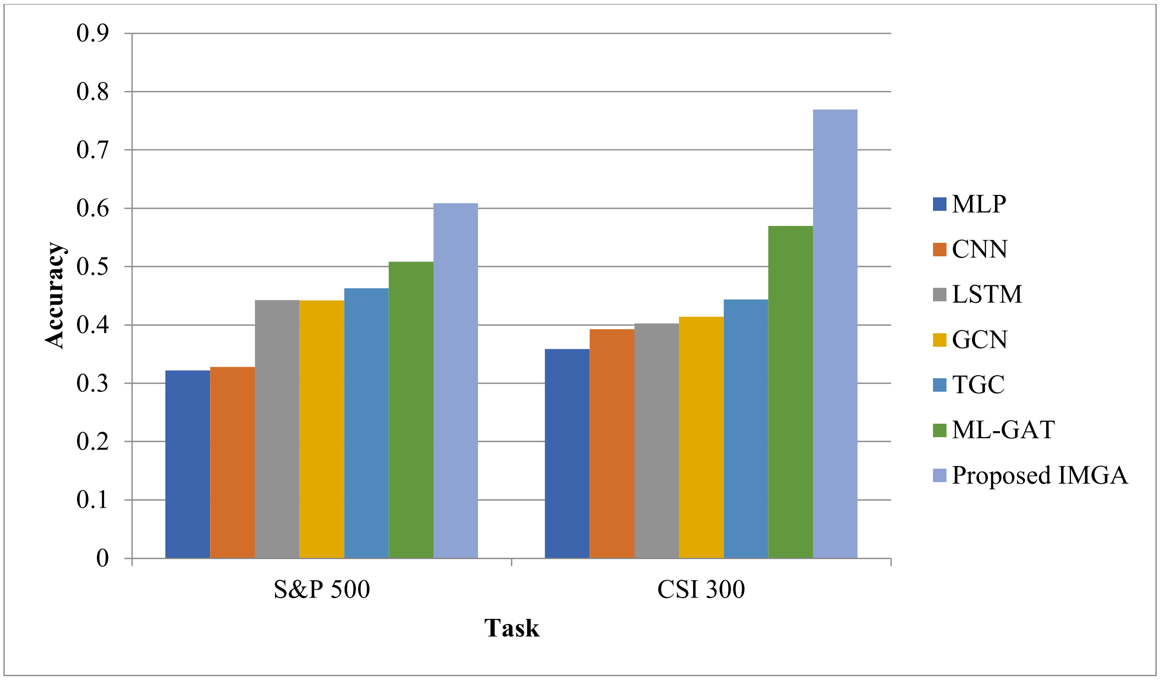

Figure 4: Classification Accuracy scores for stock prediction tasks

TABLE 7. List of the first-order Wiki relations

	Relation Index	Relation Name	Description
1	R=P127	Owend by	owner of the subject.
2	R=P155	Follows	immediately prior item in a series of which the subject is a part.
3	R=P156	Followed by	immediately following item in a series of which the subject is a part.
4	R=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization.
5	R=P361	Part of	object of which the subject is a part.
6	R=P414	Stock Exchange	exchange on which this company is traded.
7	R=P749	Parent organization	parent organization of an organization, opposite of subsidiaries.
8	R=P1830	Owner of	entities owned by the subject.
9	R=P1889	Different from	item that is different from another item, with which it is often confused

TABLE 8.List of the second-order Wiki relations.

	Relation Index	Relation Name	Description
1	R1=P17	Country	sovereign state of this item; don't use on humans
	R2=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
2	R1=P17	Country	sovereign state of this item; don't use on humans
	R2=P495	Country of origin	country of origin of this item (creative work, food, phrase, product, etc.)
3	R1=P17	Country	sovereign state of this item; don't use on humans
	R2=P740	Location of formation	location where a group or organization was formed
4	R1=P31	Instance of	that class of which this subject is a particular example and member
	R2=P452	Industry	industry of company or organization
5	R1=P31	Instance of	that class of which this subject is a particular example and member
	R2=P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process
6	R1=P31	Instance of	that class of which this subject is a particular example and member
	R2=P1454	Legal form	legal form of an organization
7	R1=P112	Founded by	founder or co-founder of this organization, religion or place
	R2=P112	Founded by	founder or co-founder of this organization, religion or place
8	R1=P112	Founded by	founder or co-founder of this organization, religion or place
	R2=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
9	R1=P121	Item operated	equipment, installation or service operated by the subject
	R2=P121	Item operated	equipment, installation or service operated by the subject
10	R1=P127	Owned by	owner of the subject
	R2=P127	Owned by	owner of the subject
11	R1=P127	Owned by	owner of the subject
	R2=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
12	R1=P127	Owned by	owner of the subject
	R2=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
13	R1=P127	Owned by	owner of the subject
	R2=P1830	Owner of	entities owned by the subject
14	R1=P131	Located in the administrative territorial entity	the item is located on the territory of the following administrative entity.
	R2=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
15	R1=P155	Follows	immediately prior item in a series of which the subject is a part
	R2=P155	Follows	immediately prior item in a series of which the subject is a part
16	R1=P155	Follows	immediately prior item in a series of which the subject is a part
	R2=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
17	R1=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
	R2=P17	Country	sovereign state of this item; don't use on humans
18	R1=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
	R2=P131	Located in the administrative territorial entity	the item is located on the territory of the following administrative entity.
19	R1=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
	R2=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
20	R1=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
	R2=P740	Location of formation	location where a group or organization was formed
21	R1=P166	Award received	award or recognition received by a person, organisation or creative work
	R2=P166	Award received	award or recognition received by a person, organisation or creative work
22	R1=P176	Manufacturer	manufacturer or producer of this product
	R2=P452	Industry	industry of company or organization
23	R1=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
	R2=P127	Owned by	owner of the subject
24	R1=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
	R2=P155	Follows	immediately prior item in a series of which the subject is a part
25	R1=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
	R2=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
26	R1=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
	R2=P1830	Owner of	entities owned by the subject
27	R1=P361	Part of	object of which the subject is a part
	R2=P361	Part of	object of which the subject is a part
28	R1=P361	Part of	object of which the subject is a part
	R2=P414	Stock Exchange	exchange on which this company is traded
29	R1=P361	Part of	object of which the subject is a part
	R2=P463	Member of	organization or club to which the subject belongs
30	R1=P414	Stock Exchange	exchange on which this company is traded
	R2=P361	Part of	object of which the subject is a part
31	R1=P452	Industry	industry of company or organization
	R2=P31	Instance of	that class of which this subject is a particular example and member
32	R1=P452	Industry	industry of company or organization
	R2=P176	Manufacturer	manufacturer or producer of this product
33	R1=P452	Industry	industry of company or organization
	R2=P452	Industry	industry of company or organization
34	R1=P452	Industry	industry of company or organization
	R2=P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process

35	R1=P452	Industry	industry of company or organization
	R2=P1454	Legal form	legal form of an organization
36	R1=P463	Member of	organization or club to which the subject belongs
	R2=P361	Part of	object of which the subject is a part
37	R1=P463	Member of	organization or club to which the subject belongs
	R2=P463	Member of	organization or club to which the subject belongs
38	R1=P495	Country of origin	country of origin of this item (creative work, food, phrase, product, etc.)
	R2=P17	Country	sovereign state of this item; don't use on humans
39	R1=P495	Country of origin	country of origin of this item (creative work, food, phrase, product, etc.)
	R2=P740	Location of formation	location where a group or organization was formed
40	R1=P625	Coordinate location	geocoordinates of the subject.
	R2=P625	Coordinate location	geocoordinates of the subject.
41	R1=P740	Location of formation	location where a group or organization was formed
	R2=P17	Country	sovereign state of this item; don't use on humans
42	R1=P740	Location of formation	location where a group or organization was formed
	R2=P159	Headquarters location	specific location where an organization's headquarters is or has been situated.
43	R1=P740	Location of formation	location where a group or organization was formed
	R2=P495	Country of origin	country of origin of this item (creative work, food, phrase, product, etc.)
44	R1=P740	Location of formation	location where a group or organization was formed
	R2=P740	Location of formation	location where a group or organization was formed
45	R1=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
	R2=P112	Founded by	founder or co-founder of this organization, religion or place
46	R1=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
	R2=P127	Owned by	owner of the subject
47	R1=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
	R2=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
48	R1=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
	R2=P1830	Owner of	entities owned by the subject
49	R1=P793	Significant event	significant or notable events associated with the subject
	R2=P793	Significant event	significant or notable events associated with the subject
50	R1=P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process
	R2=P31	Instance of	that class of which this subject is a particular example and member
51	R1=P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process
	R2=P452	Industry	industry of company or organization
52	R1=P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process
	R2=P1056	Product or material produced	material or product produced by a government agency, business, industry, facility, or process
53	R1=P1344	Participant of	event a person or an organization was/is a participant in,
	R2=P1344	Participant of	event a person or an organization was/is a participant in,
54	R1=P1454	Legal form	legal form of an organization
	R2=P31	Instance of	that class of which this subject is a particular example and member
55	R1=P1454	Legal form	legal form of an organization
	R2=P452	Industry	industry of company or organization
56	R1=P1454	Legal form	legal form of an organization
	R2=P1454	Legal form	legal form of an organization
57	R1=P1830	Owner of	entities owned by the subject
	R2=P127	Owned by	owner of the subject
58	R1=P1830	Owner of	entities owned by the subject
	R2=P355	Subsidiary	subsidiary of a company or organization, opposite of parent organization
59	R1=P1830	Owner of	entities owned by the subject
	R2=P749	Parent organization	parent organization of an organisation, opposite of subsidiaries (P355)
60	R1=P1830	Owner of	entities owned by the subject
	R2=P1830	Owner of	entities owned by the subject
61	R1=P5009	Complies with	the product or work complies with a certain norm or passes a test
	R2=P5009	Complies with	the product or work complies with a certain norm or passes a test
62	R1=P6379	Has works in the collection	collection that have works of this artist
	R2=P6379	Has works in the collection	collection that have works of this artist

IV. CONCLUSION

This paper presents a multilayer graph attention neural network model for trend prediction. To compensate for the lack of previous knowledge of existing stock forecasting algorithms, we use a specific feature extraction module to merge news, corporate relations, and financial market data into a graph neural attention network-based model. A sentiment analysis module was developed in this work using natural language processing and GRU. The findings of the BERT, LSTM, GRU, and Relations models from wikidata are fed into the factor analysis module. The newly formed IMGA was then given a set of criteria to apply in predicting stock prices.IMGA extracts relational data using GCN. IMGA tries to carefully filter various types of input to generate an aggregated graph and learn the feature representation of nodes that are useful for prediction tasks by employing numerous layers of attention mechanisms at various levels. Graph-based learning is thought to improve prediction accuracy. To assess the efficacy of the suggested approach, we contrast IMGA with widely used benchmark models built on publicly available datasets. The outcomes also highlight how crucial it is to use relational data and financial news. Different relational data aggregation techniques used by GCN, TGC, and ML-GAT result in varying prediction accuracies.

REFERENCES

[1] M. M. Rounaghi and F. N. Zadeh, ‘‘Investigation of market efficiency and financial stability between S&P 500 and London stock exchange:Monthly and yearly forecasting of time series stock returns using ARMA model,’’ Phys. A, Stat. Mech. Appl., vol. 456, pp. 10–21, Aug. 2016, doi:10.1016/j.physa.2016.03.006.

[2] G. Bandyopadhyay, ‘‘Gold price forecasting using ARIMA model,’’J. Adv. Manage. Sci., vol. 4, no. 2, pp. 117–121, 2016, doi:10.12720/joams.4.2.117-121.

[3] H. Shi, Z. You, and Z. Chen, ‘‘Analysis and prediction of Shanghai composite index by ARIMA model based on wavelet analysis,’’ J. Math. Pract.Theory, vol. 44, no. 23, pp. 66–72, 2014.

[4] H. Herwartz, ‘‘Stock return prediction under GARCH—An empirical assessment,’’ Int. J. Forecasting, vol. 33, no. 3, pp. 569–580, Jul. 2017,doi: 10.1016/j.ijforecast.2017.01.002.

[5] H. Mohammadi and L. Su, ‘‘International evidence on crude oil price dynamics: Applications of ARIMA-GARCH models,’’Energy Econ., vol. 32, no. 5, pp. 1001–1008, Sep. 2010, doi:10.1016/j.eneco.2010.04.009.

[6] A. Hossain and M. Nasser, ‘‘Recurrent support and relevance vector machines based model with application to forecasting volatility of financial returns,’’ J. Intell. Learn. Syst. Appl., vol. 3, no. 4, pp. 230–241, 2011, doi:10.4236/jilsa.2011.34026.

[7] J. Chai, J. Du, K. K. Lai, and Y. P. Lee, ‘‘A hybrid least square support vector machine model with parameters optimization for stock forecasting,’’ Math. Problems Eng., vol. 2015, pp. 1–7, Jan. 2015, doi:10.1155/2015/231394.

[8] A. Murkute and T. Sarode, ‘‘Forecasting market price of stock using artificial neural network,’’ Int. J. Comput. Appl., vol. 124, no. 12, pp. 11–15,Aug. 2015, doi: 10.5120/ijca2015905681.

[9] D. Banjade, ‘‘Forecasting Bitcoin price using artificial neural network,’’ Jan. 2020, doi: 10.2139/ssrn.3515702.

[10] J. Zahedi and M. M. Rounaghi, ‘‘Application of artificial neural network models and principal component analysis method in predicting stock prices on Tehran stock exchange,’’ Phys. A, Stat. Mech. Appl., vol. 438,pp. 178–187, Nov. 2015, doi: 10.1016/j.physa.2015.06.033.

[11] A. H. Moghaddam, M. H. Moghaddam, and M. Esfandyari, ‘‘Stock market index prediction using artificial neural network,’’ J. Econ.,FinanceAdminist. Sci., vol. 21, no. 41, pp. 89–93, Dec. 2016, doi:10.1016/j.jefas.2016.07.002.

[12] H. Liu and Y. Hou, ‘‘Application of Bayesian neural network in prediction of stock time series,’’ Comput. Eng. Appl., vol. 55, no. 12, pp. 225–229,2019.

[13] E. Hoseinzade and S. Haratizadeh, ``CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Syst. Appl.*, vol. 129,pp. 273_285, Sep. 2019.

[14] J. M.-T. Wu, Z. Li, G. Srivastava, M.-H. Tasi, and J. C.-W. Lin, ``A graph based convolutional neural network stock price prediction with leading indicators," *Softw.,Pract. Exp.*, vol. 51, no. 3, pp. 628_644, Mar. 2021.

[15] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, ``Prediction model for stock price trend based on recurrent neural network," *J. Ambient Intell. Hum. Comput.*, vol. 12, no. 1, pp. 745_753, Jan. 2021.

[16] J. M.-T.Wu, Z. Li, N. Herencsar, B. Vo, and J. C.-W. Lin, ``A graph-based CNN-LSTM stock price prediction algorithm with leading indicators," *Multimedia Syst.*, pp. 1_20, Feb. 2021

[17] A. Sherstinsky, ‘‘Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,’’ Phys. D, Nonlinear Phenomena, vol. 404, Mar. 2020, Art. no. 132306, doi: 10.1016/j.physd.2019.132306.

[18] G. Ding and L. Qin, ‘‘Study on the prediction of stock price based on the associated network model of LSTM,’’ Int. J. Mach. Learn. Cybern., vol. 11, no. 6, pp. 1307–1317, Nov. 2019, doi: 10.1007/s13042-019-01041-1.

[19] X. Yan, W. Weihan, and M. Chang, ‘‘Research on financial assets transaction prediction model based on LSTM neural network,’’ Neural Comput. Appl., vol. 33, no. 1, pp. 257–270, May 2020, doi: 10.1007/s00521-020- 04992-7.

[20] Q. Zhang, L. Yang, and F. Zhou, ``Attention enhanced long short-term memory network with multi-source heterogeneous information fusion: An application to BGI genomics," Inf. Sci., vol. 553, pp. 305-330, Apr. 2021.

[21] Y. Chen, Z.Wei, and X. Huang, ``Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in Proc. 27th ACM Int. Conf. Inf. Knowl. Manag., Oct. 2018, pp. 1655-1658.

[22] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, ``Modeling the stock relation with graph network for overnight stock movement prediction," in Proc. 29th Int. Joint Conf. Artif. Intell., Jul. 2020, pp. 4541-4547.

[23] X. Ding, Y. Zhang, T. Liu, and J. Duan, ``Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1_7.

[24] H. Wang, T. Wang, and Y. Li, ``Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 971_978.