2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom)
December, 7-9, 2023, BITS Pilani Dubai Campus, Dubai, UAE

Advancing Indian Stock Market Prediction with TRNet: A Graph Neural Network Model

Manali Patel
Department of CSE
SVNIT Surat, India
manali20101995@gmail.com

Krupa Jariwala
Department of CSE
SVNIT Surat, India
knj@coed.svnit.ac.in

Chiranjoy Chattopadhyay
School of Computing and Data Science
FLAME University Pune, India
chiranjoy.chattopadhyay@flame.edu.in

2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom) | 979-8-3503-9341-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/MoSICom59118.2023.10458729

***Abstract*—Indian stock market is unique with high volatility and complexity, so special approaches are needed for its computational understanding. This study proposes Temporal Relational Network (TRNet), a new Graph Neural Network model designed for Indian stock market prediction. TRNet considers pre-defined industry-sector links and time to predict future stock prices in NIFTY-50 index, a significant measure for Indian stock market. Using a financial knowledge graph and Graph Convolution Network, TRNet effectively captures complexities and relationships in non-Euclidean financial data. TRNet outperforms existing models in different timeframes, showing its importance for Indian stock market. We also explore how connections between stocks affect predictions and provided an explainable framework to understand the working mechanisms of the proposed approach. Combining finance and intelligent computing, TRNet paves the way for possibilities in the areas of computational finance using deep learning.**

***Index Terms*—Indian Stock market, Forecasting, Graph Neural Networks, Relational modeling, Temporal modeling, Correlation analysis, Intelligent Computing.**

I. Introduction

The stock market emerged as a top choice among other investment options as 14.3 million new investors joined the Indian stock market during the pandemic period, as stated by the official report of the State Bank of India. It offers high returns, long-term as well as short-term benefits, and protection against inflation. At the same time, stock market forecasting is a complex task due to highly volatile, non-linear, and chaotic patterns. According to the Efficient Market Hypothesis (EMH) [1], it is difficult to beat the market and earn a profit. With the recent success of Artificial Intelligence (AI) in different domains, it has also paved its way in the finance domain, which has encouraged researchers from academia and the finance domain to build an accurate prediction models [2].

The financial data exhibits temporal patterns that define the dependency of the current stock price on its previous values. Based on this, various statistical models focused on capturing temporal dependency to make predictions. These models are Auto Regressive Integrated Moving Average (ARIMA) [3], [4], Seasonal ARIMA [5], Vector Auto Regression (VAR) [6], [7], and Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) [8], [9]. But these models failed to capture the complex patterns exhibited by financial series data due to their linear and parametric nature. Recently, Deep Learning (DL) models have proven efficient in extracting features from their own and have outperformed statistical models for financial market forecasting. DL models are classified as: Artificial Neural Networks (ANNs), Convolution Neural Networks (CNNs), Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), and Generative Adversarial Networks (GANs) [10], [11]. These models focus on extracting hidden temporal patterns through their powerful sequential processing and high computing capabilities.

There are many external factors behind the volatile nature of the stock market. In literature, fundamental variables such as crude oil, gold prices, foreign exchange rates, and technical indicators built from market data are most commonly used to train deep models [12]. These factors do explain the fluctuations in stock prices, but they are not sufficient for long term predictions. These features still lack the ability to explain the variance caused by relational connections.

In the real world, stocks are not isolated entities; they are interconnected through various relationships, like ownership, supplier-consumer ties, shareholding, or sector-industry connections, which can shed light on the impact of external events. Indian stock market is unique with high volatility and complexity, so special approaches are needed. Incorporating these interconnected aspects alongside temporal dependencies enables the creation of an accurate prediction model. To address this, we propose TRNet, a deep learning-based Temporal Relational Network, designed to forecast the future price movement of companies listed on the NIFTY-50 index. To capture these relationships, we construct a financial network that considers pre-defined industry-sector linkages as an illustrative example. Utilizing Graph Convolution Network (GCN), we extract features from the non-euclidean financial network and make predictions. Through systematic experimentation, we provide a justification for the underlying principles of our proposed methodology, showcasing its efficacy in capturing stock dependencies for improved prediction.

The remaining of the article is organized as follows: Section II discusses related work. The proposed methodology is explained in Section IV. Section V represents the experimental settings in detail. Section VI represents the result and discusses the working mechanism of the proposed methodology, followed by the conclusion in Section VII.

II. Related work

From investigating traditional statistical approaches to exploring cutting-edge deep learning models, previous studies have strived to uncover the interplay of financial relationships and temporal dependencies in enhancing prediction accuracy. In this section, we review the most recent deep models utilized for stock market forecasting.

Ali et al. proposed the ANN [13] model to predict the future price movement of various indices such as the KSE-100, KOSPI, Nikkei 225, and SZSE. The proposed framework outperformed the SVM approach. Farahani and Hajiagha predicted the future price of five international indices using ANN Model [14]. To train a model various optimization algorithms were used. Srivinay et al. proposed a hybrid model using the rule ensemble method and ANN model to predict the future price of Indian banking sector stocks and it outperformed single prediction model [15]. Selvamuthu et al. utilized the ANN model [16] to predict the future movement of the Reliance company listed on the Indian stock market using tick data and 15-min data. They trained the ANN model with three different learning algorithms and found that the Scaled Conjugate Gradient approach gave the best performance.

The state-of-the-art vision-based Convolution Neural Networks (CNNs) are efficient in capturing local patterns. Inspired by this, Chen and He utilized a 1D convolution network [17] to predict the movement of the Chinese stock market. The market indicators were used to train a model in an end-to-end manner. Hiransha et al. applied deep models to predict New York and Indian stock market and found that CNN has outperformed other models [18]. Sayavong et al. applied a 1D CNN model [19] to predict the future movement of three different stocks listed on the Thai Stock Exchange. They had considered market features to train a model.

Due to their ability to process sequence data, various sequential models are suitable for stock forecasting tasks. Zhu et al. utilized the RNN model [20] to predict the future price of the Apple stock company. They experimented with two different lag values and observed that the error increased with higher lag values. This is due to the exploding and vanishing gradient problem faced by the RNN model. To eliminate that, Moghar and Hamiche utilized the LSTM model [21] to predict the future price of two different stocks listed on the New York Stock Exchange. They trained multiple layers of the LSTM model for different epochs to study its effect on the prediction results. Gao et al. studied the effect of technical indicators and investor sentiment on the prediction performance, and to select the most influencing features, two different dimensionality approaches, LASSO and PCA, were used [22]. They found that PCA combined with the LSTM gave superior performance in predicting the Shanghai Composite Index. Fathali et al. experimented with the RNN, CNN, and LSTM models to predict the future price of the NIFTY-50 index [23]. They experimented with the different feature sets to emphasize the proper feature selection. They found that the LSTM gave the best results. Maiti and Shetty predicted the future price of the five stocks listed on the NIFTY-50 using LSTM and Generative Adversarial Network (GAN), with LSTM as a generator and MLP as a discriminator. They found that the LSTM outperformed the GAN model [24].

Attention mechanism-based approaches has been successful in many sequence processing tasks. Lin et al. proposed the AT-LSTM model [25] to predict the future movement of six different indices, such as the Shanghai Stock Exchange (SSE), Shenzhen Stock Exchange (SZSE), DJIA, NASDAQ, Nikkei 225, and S&P 500 index. It outperformed LSTM, Bidirectional LSTM, and other hybrid models. Cristescu et al. incorporated the sentiment of investors to predict the future prices of the selected stocks. A VADER algorithm was used to convert the textual data into the feature vectors. These features, along with the market data, were fed into the polynomial regression model [26].

Another research direction studied the effect of interconnections existing between stock pairs to explain the variability present in the data. Instead of treating stock as an independent entity, they considered the stock market as a network [27]. Chen et al. [28] considered the shareholding ratio to build a financial knowledge graph for the companies listed in the CSI 300 index. The features extracted from the LSTM layer and adjacency matrix were given to the GCN model to perform the node classification. It also outperformed regression and sequential models. Ye et al. built multiple graphs to incorporate the relational dependency and proposed the multi-GCGRU model [29]. They applied the GCN module to extract the relational features, followed by the GRU model to consider the temporal effect. It outperformed the statistical, ANN, and LSTM models in the future movement prediction task for the CSI-300 and CSI-500 indexes. Kim et al. proposed the HATS model [30] to predict the future price movement of the stocks listed in the S&P 500 index. To enhance the prediction performance, they have considered the effect of the multiple relationships with the target firm. The historical information was encoded using the LSTM model, and the GCN model was used to update the relational embeddings. The proposed model was able to get a higher return.

From the literature, it is evident that modeling the stock market as a network yields superior results, but its effectiveness in emerging economies like India needs verification to enhance existing approaches. However, obtaining relational data poses challenges. To address this, we propose TRNet, which incorporates industry-sector data to construct a financial knowledge graph for predicting future price movements.

III. Problem Formulation

Given a feature matrix of N nodes (companies) at time instance t denoted as $X_t \in R^{N \times D}$ and an adjacency matrix A, where D denotes the number of features. Our objective is to predict the future price movement as follows:

$$y_{t+1} = f(X_t, A) \tag{1}$$

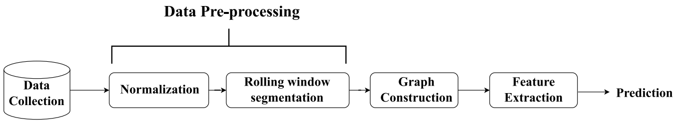

Fig. 1. A block diagram of the proposed methodology.

where $y_{t+1} \in R^{N \times 1}$ generates the future price movement of N nodes. Where '1' denotes upward trend and '-1' denotes downward trend respectively.

IV. Proposed Methodology

Figure 1 depicts the various components of the proposed methodology. In the following-subsections details of such components are described.

A. Data Pre-processing

This module performs two operations: **(1) Normalization:** to speed up the training process and **(2) Rolling Window Segmentation:** to encode the previous time steps in the current time step. These are discussed in detail below:

- **Normalization:** We have scaled the individual features with varying distributions within a uniform range of [0, 1] to boost the training process of a deep model and computed them as:

$$X_{scaled(t)} = \frac{X_t - min(X_t)}{max(X_t) - min(X_t)} \tag{2}$$

  We obtain the scaled feature matrix at time instance t $X_{scaled(t)} \in R^{N \times D}$ that consists of scaled features.
- **Rolling Window Segmentation:** The objective of rolling window segmentation is to encode a normalized input matrix $X_{scaled(t)}$ in such a manner that it represents the past T data points. The output of this layer is a time-dependent feature tensor, $\chi_t \in (T, N, D)$, where T is the lag period, N is the number of nodes (Companies), and $D = 1$ as we have considered a closing price as a feature to train a model.

B. Graph Construction

To construct a graph, we extracted the industry-sector linkage for 50 companies listed on the NIFTY-50 index from the official website of the National Stock Exchange (NSE) of India as of 2023. There are 14 different sectors listed in the NIFTY-50 index. Within those, we have merged sectors providing the same services, resulting in 10 mainstream sectors. A binary adjacency matrix $A \in N \times N$ is built by linking the stock pairs belonging to the same sector, denoted as 1 and 0 otherwise. A sample of adjacency matrix is shown in the Figure 4.

The highlighted square boxes represent different sectors and companies belonging to the same sector are connected. For an example, green square boxes represent the IT sector and all the companies belonging to this sector (Here, TCS, INFY, and TECHM) are connected. Similarly yellow and purple square boxes represent banking and automobiles sectors, respectively. In this work, we have focused on intra-sector connections.

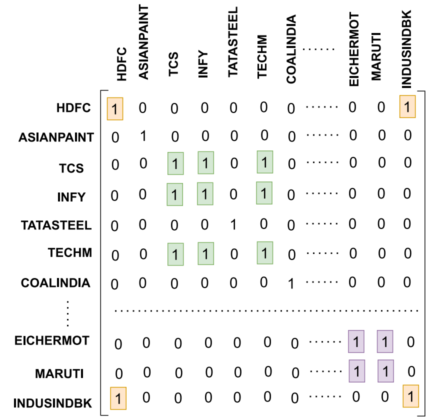

Fig. 2. Adjacency Matrix of the industry-sector graph of NIFTY-50 index.

C. Feature extraction

The objective of this module is to extract features and make a final prediction. The input to this module is the adjacency matrix A and an updated tensor χ_t constructed from the rolling window segmentation. The Graph Convolution Network (GCN) is a special version of Convolution Neural Networks that works with non-euclidean data [31]. It aggregates the feature information of neighbouring nodes and updates its hidden state accordingly. A hidden state of all nodes at layer l is updated as:

$$H^l = f(AH^{l-1}W^{l-1}) \tag{3}$$

Where A is an adjacency matrix, W is a learnable weight matrix, and $f(.)$ denotes the activation function. We have used a 2-layer GCN network with 16 and 32 units, respectively. This implies that the model will aggregate the feature information from 2-hop neighbours. For our experimentation, the equations are updated as:

$$H^1 = f(AH^0W^0) \tag{4}$$

$$H^2 = f(AH^1W^1) \tag{5}$$

where $H^0 = \chi_t$, $H^1 \in R^{N \times 16}$, $H^2 \in R^{N \times 32}$, and W^0, W^1 are learnable matrices. Extracted feature embeddings are passed to the fully connected layer for the final prediction.

V. Experiment Details

In this section, we present the experiment details.

We have considered companies listed on the NIFTY-50 index. We conducted the experiment from 2013 to 2023, which includes 2568 data points for each company. 80% is used for training, and the rest 20% is used for testing. The final dataset has 46 companies, as 2 companies had no past information about trading activities during this duration and 2 companies had no other peer companies in the NIFTY-50 index.

A. Baseline models

Comparing with a baseline model is fundamental for assessing performance, understanding the impact, ensuring generalizability, making fair comparisons, and guiding decision-making in research and practical applications. We have implemented the following deep learning models as baseline models.

- **ANN [13]:** Following [13], a multilayer model is used. It has two dense layers with 250 and 50 neurons, respectively, followed by a final dense layer.
- **1D CNN [17]:** Following [17], a 1D Convolution Neural Network is used with conv1D, Maxpooling, conv1D, Maxpooling, followed by the final prediction layer. A filter of size 3 is considered.
- **RNN [20]:** Following [20], we have implemented two different RNN layers, followed by a dense layer.
- **LSTM [21]:** Following [21], we have considered multiple layers of the LSTM model. It has two LSTM layers of units 50, followed by a final dense layer.
- **Attention-LSTM [25]:** Following [25], an attention mechanism is used. It consists of LSTM layers, followed by the attention layer and the prediction layer.

We have trained all baseline models with market data that includes Close, Open, Low, High, and Volume features.

B. Evaluation parameters

As our objective is to predict future price movements, models are evaluated based on classification parameters. These are calculated from the confusion matrix presented in Table I and listed below:

TABLE I
CONFUSION MATRIX.

		Actual values		
		Positive	Negative	Total
Predicted values	Positive	TP	FP	$TP + FP$
	Negative	FN	TN	$FN + TN$
	Total	$TP + FN$	$FP + TN$	N

- **Accuracy:** Accuracy (Equation 6) measures the fraction of predictions classified correctly by the model. It ranges from 0% to 100%.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

- **F-score:** F-score (Equation 7) It considers both precision and recall to measure a model's accuracy. The F-score measures the number of times the model has made a correct prediction on the entire dataset.

$$F - score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{7}$$

- **Matthew's correlation coefficient (MCC):** MCC (Equation 8) ranges from -1 to 1, measuring the performance of all four confusion matrix categories.

$$Matthew's\ correlation\ coefficient\ (MCC) = \frac{[TP * TN] - [FP * FN]}{\sqrt{[TP + FP] * [TP + FN] * [TN + FP] * [TN + FN]}} \tag{8}$$

The higher value of the above parameters is preferred.

VI. Result and Analysis

In this section, we report the results obtained and discuss the working mechanism of the proposed approach.

A. Experimentation Results

We have evaluated baseline and a proposed TRNet model on different lag values, i.e., 7, 15, and 30, respectively. The results are reported in Tables II-IV.

TABLE II
ANALYSIS ON LAG VALUE 7.

Model	Accuracy	F-score	MCC
ANN [13]	50.21	0.4785	0.0024
CNN [17]	49.94	0.4799	-0.0025
RNN [20]	50.39	0.4928	0.0075
LSTM [21]	50.38	0.4893	0.0069
Attention-LSTM [25]	50.27	0.4882	0.0047
TRNet	**51.44**	**0.4966**	**0.0276**

TRNet has outperformed other models with a lag value of 7. Among baseline models, the RNN model has achieved higher performance.

TABLE III
ANALYSIS ON LAG VALUE 15.

Model	Accuracy	F-score	MCC
ANN [13]	50.13	0.4787	0.0008
CNN [17]	50.10	0.4829	0.0009
RNN [20]	50.66	0.4923	0.0126
LSTM [21]	50.43	0.4862	0.0075
Attention-LSTM [25]	50.34	0.4865	0.0058
TRNet	**51.51**	**0.4987**	**0.0293**

With a lag value of 15, TRNet has outperformed other models in terms of the considered parameters. Among baseline models, RNN has given the best performance, followed by the LSTM model.

TABLE IV
ANALYSIS ON LAG VALUE 30.

Model	Accuracy	F-score	MCC
ANN [13]	50.32	0.4794	0.0045
CNN [17]	49.88	0.4791	-0.0037
RNN [20]	50.04	0.4832	-0.0002
LSTM [21]	50.57	0.4853	0.0100
Attention-LSTM [25]	50.39	0.4830	0.0063
TRNet	**51.50**	**0.4995**	**0.0290**

With a lag value of 30, TRNet has achieved the highest accuracy. Among baseline models, LSTM has achieved higher performance due to the ability to capture long-range dependency through its gating mechanism. The performance of RNN also decreases with increasing lag values (Here 30), as reported in [20]. The performance of ANN and CNN is poor for all considered lag values. Surprisingly, for all lag periods, attention-LSTM has achieved lower performance compared to the LSTM model as it requires more parameters to train. TRNet has given the best performance for short, medium and long lag values that proves the efficiency of this approach.

B. Discussion

From the analysis presented in Section VI-A, we observe that the graph-based approach TRNet that models the intra-sector associations has outperformed other baseline models. To verify the existence of interconnection, a correlation analysis is performed among companies belonging to the same sector. The correlation value defines the strength of the association between two companies. It varies from [-1,1], where 1 shows positive correlation and -1 represents negative correlation. There are 10 different sectors in which companies with large market capitalizations are classified. and their conclusive correlation analysis is depicted in Figure 3 using heatmap.

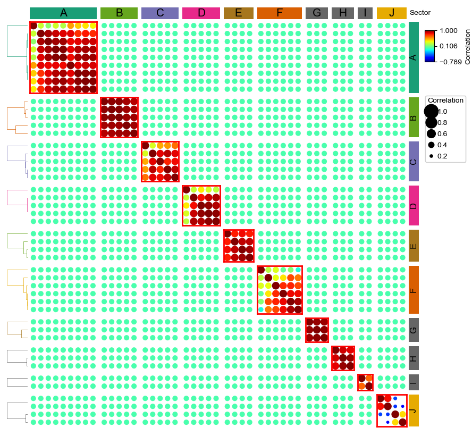

Fig. 3. Heatmap for correlation analysis of 10 sectors of NIFTY-50 index. A: Financial, B: IT, C: Healthcare, D: FMCG, E: Construction, F: Automobile, G: Metal, H: Consumer Durables, I: Power, J: Oil.

Figure 4 represents the visualization of intra-sector associations by constructing graph for each of the sector. The nodes represent individual company and edge value depicts the correlation value between them. The strong line is formed between two nodes if correlation value is greater than 0.7 and dashed line is to show correlation less than 0.7.

It is evident that there exists positive as well as negative correlation within the sectors. TRNet is efficient in capturing both the effects. To highlight the predictive power of the TRNet model over other deep learning models, accuracy is measured for companies from selective sectors with a lag value of 15. These companies are TATACONSUM (S1): FMCG, SUNPHARMA (S2): Pharmaceuticals, ICICIBANK (S3) and HDFC (S4): Banking, M&M (S5): Automobile, ASIANPAINT (S6) and TITAN (S7): Consumer Durables, WIPRO (S8): IT, LT (S9): Construction Materials, ONGC (S10) and COALINDIA (S11): Oil. The results are highlighted in Table V. Bold value indicates best performance.

For each considered company, the TRNet model outperforms other models with higher margins except the WIPRO company. This can be resolved by applying filter mechanism within the sector to study the time evolving relationship in the future. Overall considering Indian stock market as a network, predictive power is enhanced for individual companies, leading to performance improvements on the full dataset.

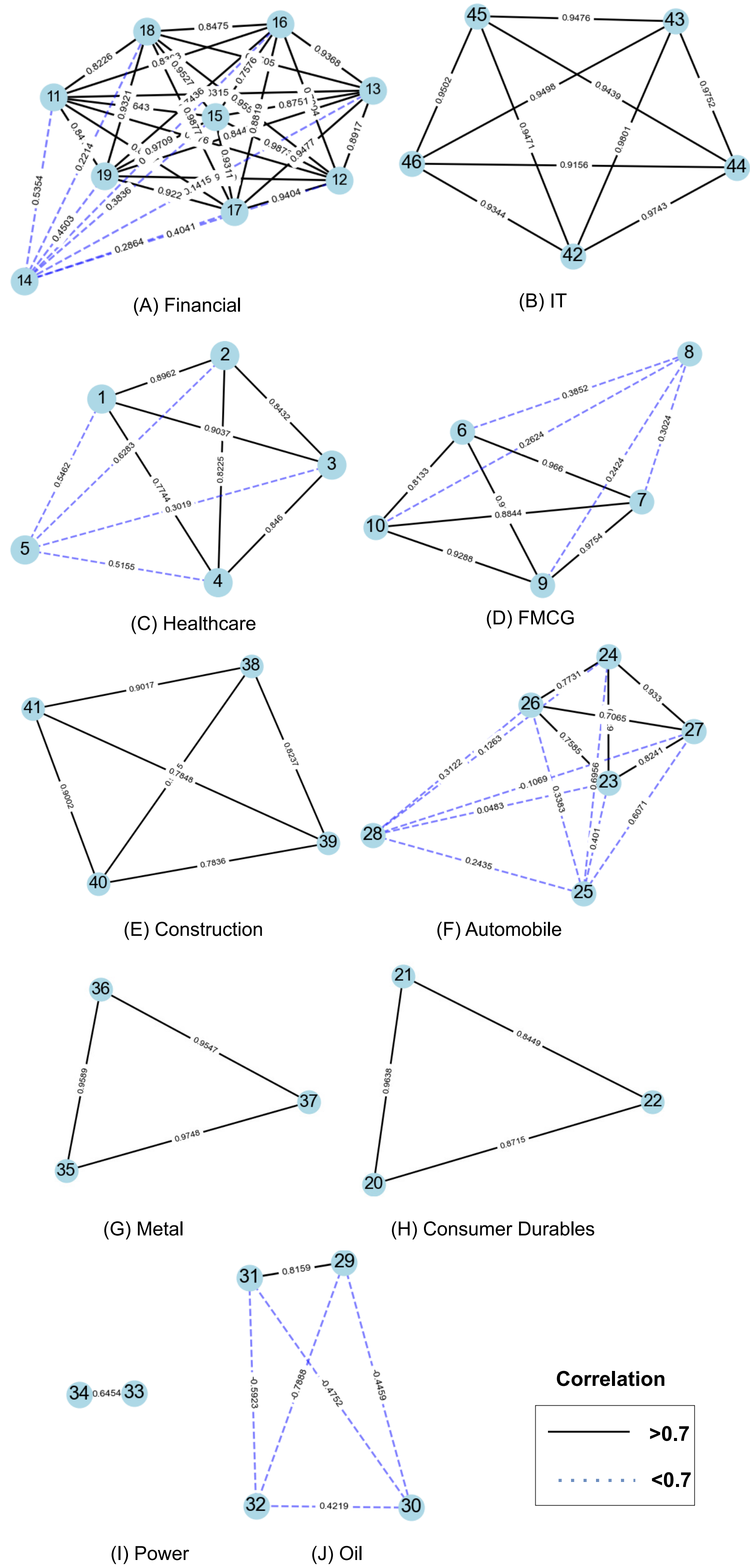

Fig. 4. Visualization of intra-sector association for 10 sectors of NIFTY-50 using graph structure.

VII. Conclusion

This work has proposed a deep learning based Temporal Relational Network (TRNet) to predict the future price

TABLE V
ACCURACY ANALYSIS ON LAG VALUE 15.

Model	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
ANN [13]	48.04	48.82	49.41	54.88	51.17	54.49	51.56	**51.17**	49.80	48.63	51.56
CNN [17]	49.41	47.07	49.60	48.04	45.89	50.19	49.80	50.58	49.02	45.89	51.56
RNN [20]	48.63	49.21	49.02	49.02	50.78	52.73	49.60	51.17	49.41	49.60	47.85
LSTM [21]	47.85	48.63	51.56	47.46	50.39	53.32	49.41	49.60	49.41	48.43	52.92
Attention-LSTM [25]	49.60	46.67	51.75	50.19	49.41	49.21	49.21	50.39	51.75	51.17	48.82
TRNet	**52.31**	**52.51**	**53.92**	**54.92**	**52.31**	**55.73**	**52.51**	50.90	**56.74**	**53.92**	**53.11**

movement of all constituent stocks of the NIFTY-50 index. In this approach, the stock market is considered an interconnected network instead of a separate entity. A financial knowledge graph is built by considering pre-defined industry-sector linkages extracted from the official website of the Indian Stock Market. From this non-Euclidean data, rich and meaningful features are extracted using the Graph Convolution Network (GCN). We have tested our approach for the different lag periods, and it has outperformed the other existing deep baseline models. An explainable framework is presented to prove the existence of relational dependency between stock pairs. The major findings suggest that incorporating relational as well as temporal dimensions can effectively explain the effect of external variables, leading to improved prediction performance for emerging economies like India. In the future, we will consider a way to study inter-sector connections, multiple relationships, and dynamic intra-sector relations to further improve the prediction accuracy.

REFERENCES

[1] E. F. Fama, “Efficient Capital Markets: A review of theory and empirical work,” The Journal of Finance, vol. 25, no. 2, p. 383, 1970.

[2] W. Lakhchini, R. Wahabi, M. Kabbouri, “Artificial Intelligence & Machine Learning in Finance: A literature review,” 2022.

[3] S. Khanderwal and D. Mohanty, “Stock price prediction using ARIMA model,” International Journal of Marketing & Human Resource Research, vol. 2, p. 98-107, 2021.

[4] J. Kowtal, “Arima Model: Forecasting The Stock Prices : A Study Conducted For Top Five Jewellery Brands In India”, J Arch. Egyptol, vol. 18, no. 08, p. 4368-4373, Jul. 2021.

[5] A. Tewari, “Forecasting NIFTY 50 benchmark Index using Seasonal ARIMA time series models,” 2020. doi: 10.13140/RG.2.2.10332.95364.

[6] N. Promma, N. Chutsagulprom N, “ Forecasting Financial and Macroeconomic Variables Using an Adaptive Parameter VAR-KF Model,” Mathematical and Computational Applications, 2023.

[7] H. Phillip, “Using Autoregressive Modelling and Machine Learning for Stock Market Prediction and Trading,” ICICT 2018, 767-774, 2018.

[8] B. Setiawan, M. Ben Abdallah, M. Fekete-Farkas, RJ. Nathan, Z. Zeman, “ GARCH (1,1) Models and Analysis of Stock Market Turmoil during COVID-19 Outbreak in an Emerging and Developed Economy,” Journal of Risk and Financial Management, vol. 14, p. 576, 2021.

[9] P. Cheteni, “Stock market volatility using GARCH models: Evidence from South Africa and China Stock Markets,” Journal of Economics and Behavioral Studies, vol. 8, no. 6(J), pp. 237–245, 2017.

[10] A. Thakkar and K. Chaudhari, “A comprehensive survey on Deep Neural Networks for stock market: The need, challenges, and future directions,” Expert Systems with Applications, vol. 177, p. 114800, 2021.

[11] M. Patel, K. Jariwala, and C. Chattopadhyay, “Deep learning techniques for stock market forecasting: Recent trends and challenges,” 2023 The 6th International Conference on Software Engineering and Information Management, 2023. doi:10.1145/3584871.3584872

[12] I.K. Nti, A.F. Adekoya and B.A. Weyori, “A systematic review of fundamental and technical analysis of stock market predictions,” Artif Intell Rev, vol. 53, p. 3007–3057, 2020.

[13] M. Ali, D. M. Khan, M. Aamir, A. Ali, and Z. Ahmad, “Predicting the direction movement of Financial Time Series using artificial neural network and support vector machine,” Complexity, pp. 1–13, 2021.

[14] M. Shahvaroughi Farahani, S.H. Razavi Hajiagha, “Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models,” Soft Comput, vol. 25, p. 8483–8513 2021.

[15] BC Srivinay Manujakshi, MG Kabadi, and N. Naik, “A Hybrid Stock Price Prediction Model Based on PRE and Deep Neural Network,” Data, 2022.

[16] D. Selvamuthu, V. Kumar, and A. Mishra, “Indian stock market prediction using artificial neural networks on Tick Data,” Financial Innovation, vol. 5, no. 1, 2019.

[17] S. Chen, H. He, “Stock Prediction Using Convolutional Neural Network,” IOP Conference Series: Materials Science and Engineering, 435, 012026, 2018.

[18] M. Hiransha, E.A. Gopalakrishnan, Vijay Krishna Menon, K.P. Soman, “NSE Stock Market Prediction Using Deep-Learning Models,” Procedia Computer Science, vol. 132, p. 1351-1362, 2018.

[19] L. Sayavong, Z. Wu, and S. Chalita, “Research on stock price prediction method based on Convolutional Neural Network,” 2019 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 2019.

[20] Y. Zhu, “Stock price prediction using the RNN model,” Journal of Physics: Conference Series, vol. 1650, p. 032103, 2020.

[21] A. Moghar and M. Hamiche, “Stock market prediction using LSTM recurrent neural network,” Procedia Computer Science, vol. 170, pp. 1168–1173, 2020.

[22] Y. Gao, R. Wang, and E. Zhou, “Stock prediction based on optimized LSTM and GRU models,” Scientific Programming, pp. 1–8, 2021.

[23] Z. Fathali, Z. Kodia, and L. Ben Said, “Stock market prediction of nifty 50 index applying machine learning techniques,” Applied Artificial Intelligence, vol. 36, no. 1, 2022.

[24] A. Maiti and P. Shetty D, “Indian stock market prediction using Deep Learning,” 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020.

[25] Y. Lin et al., “A new attention-based LSTM model for closing stock price prediction,” International Journal of Financial Engineering, vol. 09, no. 03, 2022.

[26] M. P. Cristescu, R. A. Nerisanu, D. A. Mara, and S.-V. Oprea, “Using market news sentiment analysis for stock market prediction,” Mathematics, vol. 10, no. 22, p. 4255, 2022.

[27] W. Zhang, Z. Chen, J. Miao, and X. Liu, “Research on graph neural network in stock market,” Procedia Computer Science, vol. 214, pp. 786–792, 2022.

[28] Y. Chen, Z. Wei, and X. Huang, “Incorporating corporation relationship via graph convolutional neural networks for stock price prediction,” Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018.

[29] J. Ye, J. Zhao, K. Ye, and C. Xu, “Multi-graph convolutional network for relationship-driven stock movement prediction,” 2020 25th International Conference on Pattern Recognition (ICPR), 2021.

[30] R. Kim, C. So, M. Jeong, S. Lee, J. Kim, J. Kang, Jaewoo, “HATS: A Hierarchical Graph Attention Network for Stock Movement Prediction,” 2019. arXiv preprint arXiv:1908.07999.

[31] T. N. Kipf and W. Max, “Semi-supervised classification with graph convolutional networks,” 2016. arXiv preprint arXiv:1609.02907.