Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 103–108 DOI: 10.15439/2022R02 ISSN 2300-5963 ACSIS, Vol. 33

Comparative Study of Deep Learning Models for Predicting Stock Prices

Van-Thang Duong
Faculty of Information Technology
University of Transport in Ho Chi Minh City (UTH)
Ho Chi Minh, Vietnam
thangduongvan2000@gmail.com

Duc-Tuan-Anh Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education (UTEHY)
Hung Yen, Vietnam
tuananh03@gmail.com

Thi-Thu-Hang Pham
Faculty of Information Technology
Hung Yen University of Technology and Education (UTEHY)
Hung Yen, Vietnam
thuhangpt249@gmail.com

Van-Hau Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education (UTEHY)
Hung Yen, Vietnam
haunv@utehy.edu.vn

Van-Quoc Anh Le
Faculty of Information Technology
University of Transport in Ho Chi Minh City (UTH)
Ho Chi Minh, Vietnam
anh@ut.edu.vn

***Abstract*—The stock market is volatile, dynamic, and nonlinear. Hence, predicting the stock prices has been a challenging task for any researcher in time series forecasting. Accurately predicting stock prices has been a hot topic for both financial and technical researchers. In this paper, we deploy six deep learning models (i.e., MLP, CNN, RNN, LSTM, GRU, and AE) to predict the closing price, one day ahead, of 20 different companies (i.e. 5 groups of 4) in the S&P 500 markets over the 7-years range (Jan 2015 - August 2022). The experimental results do not provide interesting insights, but also help us to deepen our understanding of how to use deep learning models in financial markets.**

***Index Terms*—stock price prediction, deep learning, machine learning.**

I. Introduction

The stock market plays an important role in the global economy today. Accurately predicting stock prices can avoid investment risk and lead to a hugely profitable investment. Recently, it has been a hot topic for both financial and technical researchers [1]–[3]. In fact, the stock market is volatile, dynamic, and nonlinear. Hence, predicting the stock prices has been a challenging task for any researcher in time series forecasting due to various factors, such as the global economy, political conditions, the company's financial performance, etc.

There are two approaches to predicting the stock markets. The first approach is qualitative (or fundamental analysis) [1], in which the intrinsic values are examined, such as market situation, financial factors, management effectiveness, consumer behaviors, and information from social media, economic analyst, etc.[1] The second approach is the technical analysis, in which the historical stock market activities are examined, for example, the price of opening, closing, maximum, minimum, adjusted closing prices and volume of a day.

Unlike the fundamental analysis that is useful for long-term investment, the technical analysis is easily influenced by short-term news. The second approach consists of two main methods, the traditional method and the machine learning method. To predict stock prices, the former widely uses classical techniques for time series models, like AR(Autoregressive), MA(Moving Average), ARMA(Autoregressive Moving Average), and ARIMA (Autoregressive Integrated Moving Average). In the latter, i.e. the machine learning method, while classical machine learning-based models have been extensively studied and obtained significant results due to their effectiveness despite working with a limited amount of data, deep learning-based models, nevertheless requiring a huge amount of data is gaining numerous interests because of its strength of learning complex patterns in unstructured data.

In this paper, we aim at taking advantage of deep learning models to predict stock prices. Our work has three contributions:

- We carry out experiments to evaluate 6 deep learning models.
- We conduct experiments on various stock prices (i.e. 5 five sectors of 4 stock prices) of S&P 500 databases.

[1] https://www.investopedia.com/terms/f/fundamentalanalysis.asp

- We contribute some insights that might help us to deepen our understanding on how to use deep learning models.

The paper is organized as follows. Section II provides the related works. Section III presents 6 deep learning models. Data and the measurements are presented in Section IV. Then, the experimental results are provided in Section V. Finally, we conclude in Section VI.

II. Related Works

The strength of deep learning models is able to find hidden features (or patterns) through a self-learning mechanism. The challenge of these models is to require a massive amount of data. However, we can collect lots of data on the stock markets easily. In this part, we will provide some works which are related to exploiting deep learning models for predicting the stock market.

To predict stock behavior, by collecting Google domestic trends as indicators. Xiong et al. [4] compare a traditional model for time series data, namely the GARCH (Generalized Autoregressive Conditional Heteroskedasticity) [5], and a state-of-the-art model for dealing with long-term dependencies, namely LSTM (Long Short-Term Memory) [6]. The results show that LSTM is superior to GARCH. Fischer and Krauss [7] deploy LSTM to the stocks of the S&P 500 from 1992 until 2015. Interestingly, LSTM performs better than a random forest, a standard deep neural net, and a logistic regression. Yu and Yan [8] combine LSTM with the time series phase-space reconstruction (PSR) method to predict stock prices. The experiments performed on six stock indices for various markets (S&P 500, DJIA, N 225, HSI, CSI 300, and ChiNext) demonstrate that the proposed model outperforms ARIMA, SVR (Support Vector Regressor), MLP (Multilayer Perception), and LSTM without PSR. Furthermore, Karmiani et al. [9] compare several predictive algorithms for the stock market. The results conducted on nine technology companies show that LSTM is a better choice compared to SVM, Backpropagation, and Kalman filter algorithms. To predict the index price of the stock market on the next day, Gao et al. [10] evaluate MLP, LSTM, CNN (Convolutional Neural Network), and attention-based neural network. To do so, S&P 500 index (the most developed market), CSI 300 index (the less developed market), and Nikkei 225 index (the developing market) are considered. The authors show that the attention-based model is insignificant better than the three other models.

It is reasonable to understand that LSTM has gained significant popularity in stock prediction because LSTM is a state-of-the-art model to deal with sequential data. Surprisingly, to the best of our knowledge, there are a few attempts to employ deep learning models, including LSTM, in financial markets. Hence, it inspires us to experimentally conduct the comparison between deep learning models in terms of predicting stock prices.

III. Models

Here we present 6 typical deep learning models which are used to conduct the experiment results in Section V.

A. Multilayer Perceptron (MLP)

An artificial Neural Network (ANN) is a computational model which imitates the way of information processing by neurons in the human brain by making the right connections among nodes. An ANN consists of three parts—a layer of input nodes, layers of hidden nodes, and finally a layer of output nodes (see Figure A1). Each layer consists of a group of multiple neurons/nodes which are connected to others via weighted links. A multilayer perceptron (MLP) is a fully connected network of ANN. The input layer processes the input data and passes it to the hidden layer, then the hidden layer handles the outcome from the previous layer and passes it to the next layer, finally, the output layer produces the result. The learning capability of MLPs takes place by way of adjusting weight values. Thanks to their well-organized structures, MLPs are able to successfully map nonlinear input to output by automatically extracting subtle patterns and multiple features from a large dataset through each layer (see Figure 1).

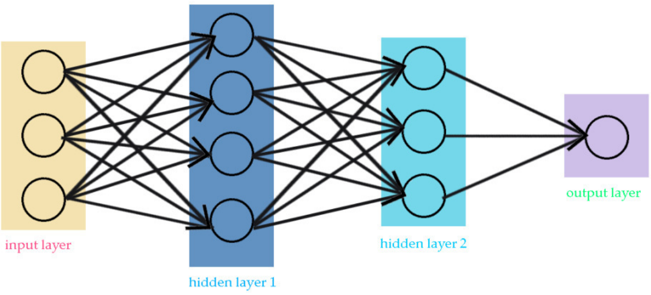

Fig. 1. An illustration of a simple MLP which consists of two consecutive hidden layers located between the input layer and the output layer. every neuron is interconnected and assigned weights (represented by arrows). Each neuron learns/adjusts its weights through its inputs and desired outputs.

B. Convolutional Neural Network (CNN)

The breakthrough in MLP happened when one variant of the models—Deep Convolutional Neural Networks—was ranked first in the ImageNet Large Scale Visual Recognition Challenge after approaching human performance in image classification [11]. Therefrom, Convolutional Neural Networks (CNN) and their descendants have been approaching superhuman performance in a wide range of domains, including pattern recognition, natural language processing, video processing, speech recognition, and time-series forecasting [12].

C. Recurrent Neural Network (RNN)

Notwithstanding the extreme success, CNNs face a big issue: they are not able to cope with time series and sequential data. In order to deal with this issue, Recurrent Neural Network [13]—one of the deep neural network models—was introduced. Due to the flexibility in architecture, computational power, and the rich inherent memory through feedback, RNNs have a wide range of applications in sequential data, including machine translation [14], speech recognition [15], time series anomaly detection [16], and time series forecasting [17].

TABLE I
List of twenty stocks divided into five groups. The symbol of stocks and their market cap are shown. T stands for trillion and B denotes billion.

Communication Services		Consumer Cyclical		Energy		Healthcare		Technology	
Stock	Market Cap	Stock	Market Cap	Stock	Market Cap	Stock	Market Cap	Stock	Market Cap
GOOG	1.397T	AMZN	1.285T	XOM	395.717B	UNH	484.229B	AAPL	2.483T
META	426.084B	TSLA	859.887B	CVX	307.551B	JNJ	429.03B	MSFT	1.889T
DIS	200.701B	HD	296.411B	SHEL	195.57B	LLY	291.533B	TSM	410.636B
VZ	172.608B	BABA	234.856B	TTE	132.514B	PFE	256.821B	NVDA	335.548B

D. Long Short-Term Memory (LSTM)

Nevertheless, RNNs have two serious problems: vanishing/exploding gradients and learning with long term dependencies [6]. First, like any CNNs, RNNs are able to learn by adjusting weight values. Technically, weight values are updated through the backpropagation algorithm. Unfortunately, in the case of RNNs with a large number of hidden layers, performing the backpropagation algorithm leads to vanishing gradient (i.e, exponential decrease) and exploding gradient (i.e., exponential growth) problems because a large number of derivatives have to be multiplied. Second, RNNs are only able to capture short-term dependencies in sequences. Therefore, we need a new type of architecture design to deal with the two above problems affecting RNNs. Fortunately, Long Short-Term Memory (LSTM) was introduced by Hochreiter and Schmidhuber [6]. LSTM is a type of RNN and is specifically designed to deal with longer dependencies in sequences [18] and reduce the exploding gradients. Unlike RNNs, instead of adding regular neural units (i.e., hidden layers), LSTM adds memory blocks. A common LSTM memory block consists of a cell state and three gates—an input gate, a forget gate, and an output gate.

E. Gated Recurrent Unit (GRU)

To deal with the vanishing gradient and exploding gradient problems, Cho et al. [19] introduced the gated recurrent unit (GRU) which is a variation of LSTM in terms of the architecture designed without the output gate. While LSTM consists of three gates (i.e., input, forget, and output gates), GRU comprises two gates: the reset gate and the update gate. Therefore, GRU has fewer parameters. The advantage of GRN over LSTM is better computation, although GRN obtains comparable results in many cases compared to LSTM [20].

F. Autoencoder (AE)

Autoencoder (AE) is a special artificial neural network [21]. AE consists of three parts: 1) encoder that converts the input into the bottleneck; 2) bottleneck that is a compressed representation keeping only the most important information; and 3) decoder that reconstructs the original input from the bottleneck. In fact, AE is able to handle data in which the features are correlations. Hence, dealing with noisy data is an interesting advantage of Autoencoder.

IV. Data and Evaluation

A. Data

In order to evaluate six deep learning models, we perform them on datasets with different characteristics. Particularly, there are five groups of stocks corresponding to five sectors where companies are working. They are Consumer Cyclical, Communication Services, Energy, Healthcare, and Technology. For each group, we select four stocks of influential companies in their sector, list of 20 stocks is shown in Table I.

The historical data of each stock is collect from January 2^{nd} 2015 to August 8^{th} 2022. There are six columns in one dataset: Date, Open, High, Low, Close, Volume. Figure 2 presents data of CVX stock from group Energy, Chervon Corporation engages in integrated energy and chemicals operations worldwide.

B. Evaluation

The performances of six models were evaluated using five statistical indices, i.e., Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Normalized Root Mean Squared Error (NRMSE) and Coefficient of Determination (R2), which can be expressed as follows:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{2}$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{y_i}| \tag{3}$$

$$NRMSE = \frac{100 \times RMSE}{y_{max} - y_{min}} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{5}$$

where $\hat{y}_i$ is the i^{th} predicted daily closing price, y_i is the i^{th} objected daily closing price, $\bar{y}$ denote the mean observed daily closing price and n is the total number of data samples evaluated.

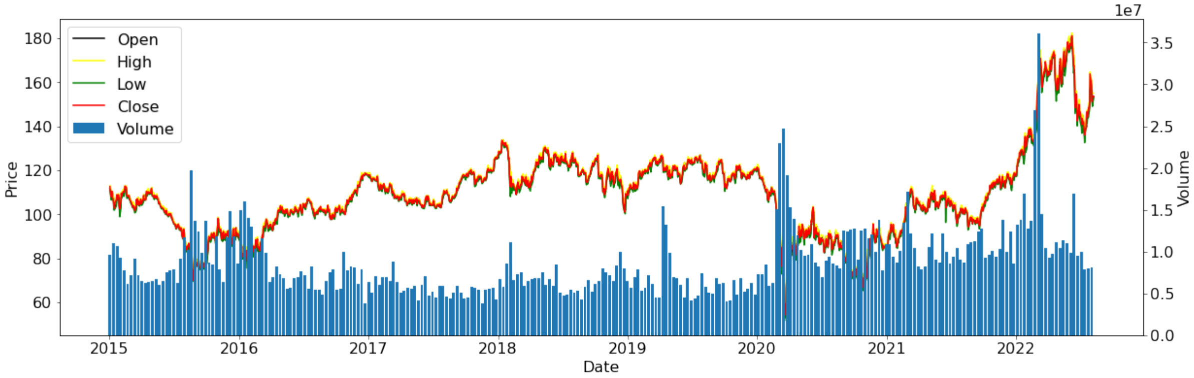

Fig. 2. Historical prices of CVX stock collected from January 2^{nd} 2015 to August 8^{th} 2022. The volumes of stock are presented by bar chart and four line charts respectively present Open, High, Low, Close prices of CVX stock.

V. Experimental Results

In this paper, we use Close price of the next day as target variable, while five features including Open, High, Low, Close, Volume are chosen as input. In forecasting stock problems, finding look-back window's size (lags) is an important part. However there isn't a right thesis about choosing the best lags of time for a forecasting problem. Therefore we run experiment to find out the best input size for forecasting stock problems. We run models in five cases with different input sizes consists of 3 days, 1 week, 2 weeks, 3 weeks and 1 month input. Overall, we run experiment for six deep learning models on twenty stocks, the result in one stock is summarized by five cases with different input sizes. The results of experiment allow us to compare performance of models and also select best input to predict stock prices.

TABLE II
Performance results of six models for forecasting closing prices of CVX stock using 3 days of input.

	MAE	RMSE	MAPE	NRMSE	R2
MLP	2.0789	2.8744	0.0158	2.9964	0.9878
RNN	**1.8899**	**2.6486**	**0.0146**	**2.7610**	**0.9897**
LSTM	2.1624	3.1518	0.0167	3.2856	0.9854
GRU	2.0294	*2.8022*	0.0156	*2.9211*	*0.9884*
CNN	2.1854	2.9568	0.0168	3.0823	0.9871
AE	*2.0041*	2.8204	*0.0153*	2.9401	0.9883

Table II presents the performance results between models for predicting CVX stock's price with lag time equal to 3 days. The table shows that forecast results of the RNN model are more accurate than other models. The RNN model has MAE of 1.899, RMSE of 2.649, MAPE of 0.015 and NRMSE of 2.761, which are smaller than all of the other models. For all of six models, the R2 indices have values approximately to 0.99 indicate a good correlation between the observed and forecasted values.

On a different dataset and a different input size, the results of models are changed. The forecast results of six models for UNH stock by using 1 week of input are shown in Table III.

TABLE III
Performance results of six models for forecasting closing prices of UNH stock using 1 week of input.

	MAE	RMSE	MAPE	NRMSE	R2
MLP	13.2710	15.9347	2.8767	7.1885	0.9204
RNN	6.7058	8.6229	1.4866	3.8900	0.9767
LSTM	7.0727	9.3069	1.5539	4.1985	0.9728
GRU	17.2761	19.4373	3.7737	8.7686	0.8816
CNN	**5.3518**	**6.9854**	**1.1976**	**3.1512**	**0.9847**
AE	10.7642	13.2524	2.3403	5.9784	0.9449

By comparing performance results of models, we can see that the CNN model is the best model in this case.

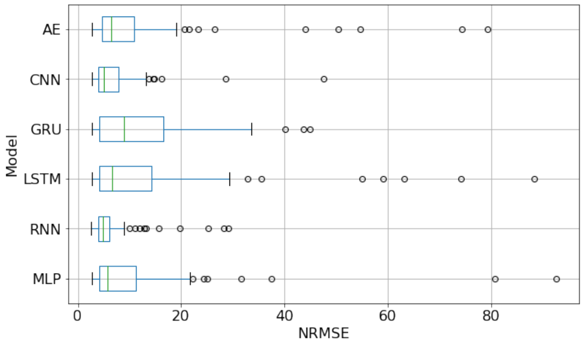

Fig. 3. Boxplot results of six models for predicting stock prices in all cases of experiment using NRMSE indices.

Looking at Table II and III, the difference between MAE and RMSE indices from these tables are quite big. The MAE indices in Table II range from 1.89 to 2.18 and in Table III they have values from 6.71 to 17.28. For RMSE indices, Table II shows results from 2.65 to 3.15 and Table III shows results from 6.99 to 15.93. Look at equation (1) and (2) we can know the reason of these differences. Two equations (1), (2) use the absolute error between the observed and forecasted values, while price of stocks are different, the results on two stocks are also different using MAE and RMSE indices. In order to summarize results from multiple stocks, we use three indices:

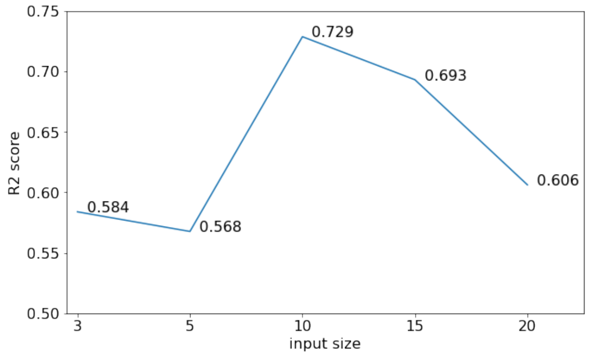

Fig. 4. Results of models in all cases based on the input size. The mean of R^2 indices are reported.

MAPE, NRMSE and R^2. These three indices calculate relative error between real and predicted values, so we can compare the results of models between stocks and discover in a bigger picture.

Figure 3 provides the boxplot of NRMSE results for six models in all cases of experiment. From the boxplot, we can compare the performance of models in forecasting stock problems. Through the median NRMSE of the RNN model is not significantly lower than the CNN model, the width of the box and the values of the right boundaries are significantly smaller in the RNN model than in the other five models, indicating that the RNN model provides an overall lower forecast error.

The R^2 or coefficient of determination, provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the models. The larger R^2 score, the better result and the best result appears when R^2 score equal to 1. In Figure 4, we don't compare the performance of models together but summarize the results from models and group them by each sizes of input. As we can see, changing input size from 3 days to 5 days doesn't improve the results too much. When the input size equal to 10 days e.g. 2 week, the R^2 score is 0.729 and is the largest score in 5 cases of input sizes. Continue increasing input size to 15 and 20 days make the forecast results become worse because the R^2 scores decrease.

From Figure 3 and 4, we can answer the question: which are the best model and the best input size for the predicting stock price problems? Figure 3 shows that the RNN model is the best model overall and Figure 4 tells us that 10 days is the best size of input for these problems. Now we compare models in a smaller case - group of stocks and find the best model for each group. Figure 5 presents the performance of models using MAPE indices. There are 5 groups: Communication Services, Consumer Cyclical, Energy, Healthcare and Technology, for each group we show results of six models. In Energy group, the performance of six models are all good and GRU model has smallest forecast errors. In the 4 other groups, the results of the RNN model and the CNN model outperform the results of the 4 other models. Particularly, the RNN model has best result and the second is the CNN model in Communication Services and Technology group. In Healthcare and Consumer Cyclical group, the RNN model is the second and the CNN model has the best performance.

VI. Conclusions

Through the paper, we have observed that the RNN model performs the most accurate results in terms of predicting stock prices. However, the choice of models depends on the sector we want to predict. If we want to predict the sector of Communication Services and Technology, the RNN model is the best choice. In the sector of Consumer Cyclical and Healthcare, the best model is the CNN model. And we should use the GRU model for predicting stocks in the Energy sector. Unlike many previous studies that demonstrated that the LSTM model is a state-of-the-art model to deal with financial time series data, our results show that the RNN model is the most suitable model for predicting stock prices because of the short temporal dependency of data.

In the future, we are going to exploit several more deep learning models (e.g., Variational Autoencoder - VAE, Generative Adversarial Network - GAN) on further companies.

References

[1] J. Hur, M. Raj, and Y. E. Riyanto, "Finance and trade: A cross-country empirical analysis on the impact of financial development and asset tangibility on international trade," *World Development*, vol. 34, no. 10, pp. 1728–1741, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305750X06001148

[2] L. Li, Y. Wu, Y. Ou, Q. Li, Y. Zhou, and D. Chen, "Research on machine learning algorithms and feature extraction for time series," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–5.

[3] "Forecasting directional movements of stock prices for intraday trading using lstm and random forests," *Finance Research Letters*, vol. 46, p. 102280, 2022.

[4] R. Xiong, E. P. Nichols, and Y. Shen, "Deep learning stock volatility with google domestic trends," 2015. [Online]. Available: https://arxiv.org/abs/1512.04916

[5] R. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982. [Online]. Available: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:50:y:1982:i:4:p:987-1007

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[7] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221717310652

[8] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1609–1628, 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04212-x

[9] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 228–234.

[10] P. Gao, R. Zhang, and X. Yang, "The application of stock index price prediction with neural network," *Mathematical and Computational Applications*, vol. 25, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2297-8747/25/3/53

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc.

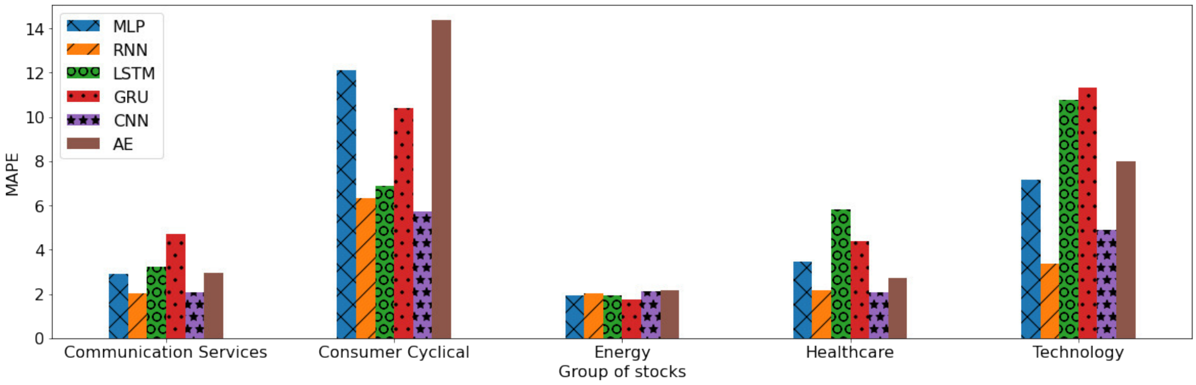

Fig. 5. Performance results of six models divided into five groups. The results are summarized by the mean of MAPE indices.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *CoRR*, vol. abs/1404.7828, 2014. [Online]. Available: http://arxiv.org/abs/1404.7828

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*. Cambridge, MA, USA: MIT Press, 1988, p. 696–699.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[15] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005, iJCNN 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608005001206

[16] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *ESANN*, 2015.

[17] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207020300996

[18] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Technical University Munich, 2008. [Online]. Available: https://d-nb.info/99115827X

[19] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014. [Online]. Available: http://arxiv.org/abs/1409.1259

[20] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[21] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.