DELAFO: An Efficient Portfolio Optimization Using Deep Neural Networks

Hieu K. Cao[1,3], Han K. Cao[2], and Binh T. Nguyen[1,2,4,5](✉)

[1] AISIA Research Lab, Ho Chi Minh City, Vietnam
[2] Inspectorio Research Lab, Ho Chi Minh City, Vietnam
[3] John Von Neumann Institute, Ho Chi Minh City, Vietnam
[4] University of Science, Ho Chi Minh City, Vietnam
ngtbinh@hcmus.edu.vn
[5] Vietnam National University in Ho Chi Minh City, Ho Chi Minh City, Vietnam

Abstract. Portfolio optimization has been broadly investigated during the last decades and had a lot of applications in finance and economics. In this paper, we study the portfolio optimization problem in the Vietnamese stock market by using deep-learning methodologies and one dataset collected from the Ho Chi Minh City Stock Exchange (VN-HOSE) from the beginning of the year 2013 to the middle of the year 2019. We aim to construct an efficient algorithm that can find the portfolio having the highest Sharpe ratio in the next coming weeks. To overcome this challenge, we propose a novel loss function and transform the original problem into a supervised problem. The input data can be determined as a 3D tensor, while the predicted output is the unnormalized weighted proportion for each ticker in the portfolio to maximize the daily return Y of the stock market after a given number of days. We compare different deep learning models, including Residual Networks (ResNet), Long short-term memory (LSTM), Gated Recurrent Unit (GRU), Self-Attention (SA), Additive Attention (AA), and various combinations: SA + LSTM, SA + GRU, AA + LSTM, and AA + GRU. The experimental results show that the AA + GRU outperforms the rest of the methods on the Sharpe ratio and provides promising results for the portfolio optimization problem not only in Vietnam but also in other countries.

Keywords: Portfolio optimization · Self-attention · Addictive attention · Residual Network · LSTM

1 Introduction

Using historical stock data for portfolio optimization has been one of the most exciting and challenging topics for investors in the financial market during the

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-47426-3_48) contains supplementary material, which is available to authorized users.

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12084, pp. 623–635, 2020.
https://doi.org/10.1007/978-3-030-47426-3_48

last decades [1,2]. Many factors have different influences on the stock price, and it is essential to extract a list of crucial factors from both historical stock prices and other data sources. As there is no such thing as a free lunch, investors have to find an efficient strategy for a trade-off between getting more profits and reducing the investment risk. Sometimes, they need to invest multiple assets for diversifying the portfolio.

Traditionally, one can use statistical methods for predicting a financial time series problem. There are popular techniques, including autoregressive moving average (ARMA) [3], autoregressive conditional heteroscedastic (ARCH) [4], and autoregressive integrated moving average (ARIMA) [5]. Importantly, these statistical methods usually consider the stock time series as a linear process and then model the generation process for a latent time series to foresee future stock prices. Practically, a stock time series is generally a nonlinear dynamic process. There are many different approaches, including artificial neural networks (ANN), support vector machines (SVM), and other ensemble methods [6] to capture nonlinear characters from a given dataset without knowing any prior information. Especially, deep neural networks such as e.g. convolutional neural networks (CNN) and recurrent neural networks (RNN) have been proven to work well in many applications and multi-variable time series data.

The future price represents the future growth of each company in the stock market. Typically, the stock price of each company listed in a stock market can vary whenever one puts a sell or buy order, and the corresponding transaction completes. Many factors have influenced the stock price of one company, for example, such as the company's net profit, demand stability, competitive strength in the market, new technology used, and production volume. Also, the macro-economic condition can play a unique role in the stock market as well as the currency exchange rate and the change of the government's policies. After boasting increased macro-economic stability and improving the pro-business financial environment, Vietnam has become one of the world's most attractive markets for international investors. With the population of nearly 100 million people and most of whom are young people (under the age of 35), Vietnam can provide a young, motivated, highly skilled, and educated workforce to multiple international startups and enterprises with a competitive cost. At the moment, Vietnam's stock exchange is considered as one of the most promising and prospective market in the Southeast Asia. Especially, the Ho Chi Minh Stock Exchange (HOSE)[1] is becoming one of the largest securities firms in terms of both capital and size. Since launching in 2002, it has been performing strongly and more and more investors continue exhibiting a special interest in both Vietnam stock market. HOSE is currently predicted to be upgraded to an emerging market in 2021.

Up to now, there have existed a large number of useful applications using machine learning techniques in different aspects of daily life. Duy and co-workers combine deep neural networks and Gaussian mixture models for extracting brain tissues from high-resolution magnetic resonance images [7]. Deep neural networks

[1] https://www.hsx.vn/.

can also be applied to automatic music generation [8], food recognition [9], and portfolio optimization problem [10–12]. In this paper, we aim at investigating a portfolio optimization problem in which by using historical stock data of different tickers, one wants to find the equally weighted portfolio having the highest Sharpe ratio [13] in the future. This is one of the winning solutions in a well-known data science competition using the HOSE stock data in 2019. In this competition, one can use one training dataset, including all volume and prices of different tickers appearing in the Vietnam stock market from the beginning of the year 2013 to the middle of the year 2019 (July 2019), for learning an appropriate model of the portfolio optimization problem. It is worth noting that the Sharpe ratio is often used as a measure of the health of a portfolio. One usually expects that the higher the Sharpe ratio of one portfolio is in the past, the larger its Sharpe ratio is in the future. In this work, we assume that there are no new tickers joining the stock market during the testing period.

We study the portfolio optimization problem by assuming that there are N tickers in the stock market, only using the historical stock data during the last M days for training or updating the proposed model, and then doing prediction for the equally weighted portfolio having the highest Sharpe ratio during the next K days. Different from other approaches using statistical methods or time series algorithms, we transform the input data into a 3D tensor and then consider each input data as an image. As a result, we have a chance to apply different state-of-the-art methods such as e.g. Residual Networks (ResNet) [14], Long-short term memory(LSTM) [15], Gate Recurrent Unit [16], Self-Attention (SA) [17], and Additive Attention (AA) [18] for extracting important features as well as learning an appropriate model. Also, we compare them with different combinations of these techniques (SA + LSTM, SA + GRU, AA + LSTM, and AA + GRU) and measure the actual performance in the testing dataset. The experimental results show that the AA + LSTM outperforms other techniques in terms of achieving a much better value of the Sharpe ratio and a comparably smaller value of the corresponding standard deviation.

2 DELAFO: A New DeEp Learning Approach for portFolio Optimization

In this section, we present our methods to solve the portfolio optimization using the VN-HOSE dataset and deep neural networks.

2.1 Problem Formulation

We consider a dataset collected from the Vietnamese stock market from the beginning date D_0 and the ending date D_1 and N is the number of tickers appearing during that period of time. We denote $T = \{T_1, T_2, .., T_N\}$ as the list of all tickers in the market during the time window. For a given ticker T_i , $v_{i,j}$ and $p_{i,j}$ are the corresponding volume and price on the day d_j, consecutively. Moreover, we assume that all investors aim to determine the list of potential tickers in their portfolio for the next K days without putting any weight for different tickers (or equally weighted, such as e.g. $1/N$). It is important to note that all the investors usually do not want their portfolios to have a few tickers or "put all the eggs in one bucket" or lack of diversity. Having too many tickers may cost a lot of management time and fee as well. As a consequence, the outcome of the problem can be regarded as an N-binary vector (N is the number of tickers), where a one-valued entry means the corresponding ticker is chosen; otherwise, it is not selected. There are two main constraints in this problem: a) Having the same proportion for each tickers in portfolios; (b) The maximum numbers of tickers selected is 50.

The *daily return*, $R_{i,j}$, of the ticker T_i at the day d_j can be defined as $R_{i,j} = p_{i,j}/p_{i,j-1} - 1$ for all $i = 1, \ldots, N$ [13]. The daily return of portfolio can be computed by $R = \sum_1^N w_i * R_i$, where $\sum_{i=1}^N w_i = 1$. In the equally weighted portfolio optimization problem, one can assume that $w_i = \frac{1}{N}$, and therefore, the "Sharpe ratio" can be determined as [13]:

$$\text{Sharpe Ratio} = \sqrt{n} * \frac{\mathbb{E}[R - R_f]}{\sqrt{var[R - R_f]}}, \tag{1}$$

where n is an annualization factor of period (e.g, n= 252 for trading date in one year) and R_f is the risk-free rate, the expected return of any portfolio with no risk. In this work, we choose $R_f = 0$. Combining with $\mu = \frac{\sum_i^n R_i}{n}$ and $Q = \frac{\sum_i^n (R_i - \mu)^2}{n-1}$, the Sharpe ratio is calculated as:

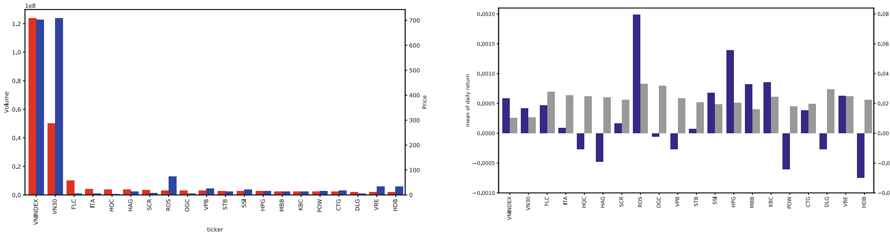
$$\text{Sharpe Ratio} = \sqrt{n} * \frac{\mathbb{E}[R]}{\sqrt{var[R]}} = \sqrt{n} * \frac{\hat{w}^T \mu}{\sqrt{\hat{w}^T \mathbf{Q} \hat{w}}}, \tag{2}$$

where $\hat{w}$ is an estimated preference vector for the list of all tickers. Typically, the equally weighted portfolio optimization problem can be formulated as follows:

$$\begin{aligned} \text{minimize } f(w) &= -\frac{w^T \mu}{\sqrt{w^T \mathbf{Q} w}}, \\ \text{subject to: } g(w) &= \mathbb{1}^T . w \leq N_0, \\ w_i &\in \{0, 1\}, \forall i = 1, \ldots, N. \end{aligned} \tag{3}$$

Here, N_0 is the maximum number of tickers selected in the optimal portfolio ($N_0 = 50$ in our initial assumption). The main goal in our work is to estimate the optimal solution w^{opt} for the portfolio optimization problem (3) in order to obtain the maximum Sharpe ratio during the next K days.

To study a deep learning model for the portfolio optimization problem, we aim at only using the historical stock data during the last M days for training and then predict the optimal equally weighted portfolio having the highest Sharpe ratio during the next K days. To solve the optimization problem in (3), we represent each input data as a 3D tensor of size $N \times M \times 2$ that includes all stock data (both the volume and the price) of N different tickers during the



(a) The total volume and price

(b) The mean and the standard deviation of the daily return

Fig. 1. Top 20 tickers in VN-HOSE.

last M consecutive days and the corresponding output of the deep network is a vector $\hat{w} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_N)$ of size $N \times 1$, where $\hat{w}_i \in [0, 1]$ $(i = 1, \ldots, N)$ is the estimated preference rate of the i-th ticker in the list of all N tickers in the stock market. Finally, the optimal portfolio can be determined by the corresponding estimated solution w^{opt}, where the i-th ticker can be chosen if the corresponding preference $\hat{w}_i \geq \theta$ (or $w_i^{opt} = 1$); otherwise, it is not selected.

2.2 A New Loss Function for the Sharpe-Ratio Maximization

Traditionally, one can estimate the maximum value of the Sharpe ratio by solving the following optimization problem [19] :

$$\hat{w} = \text{argmin} \left(w^T Q w - \lambda w^T \mu\right) / \left(w^T w\right). \qquad (4)$$

Although it does not directly optimize the Sharpe ratio as shown in Eq. (2), one can use the stochastic gradient descent method for approximating the optimal $\hat{w}$ [20]. In this paper, we propose the following new loss function for the equally weighted portfolio optimization problem:

$$L(\hat{w}) = -\left(\hat{w}^T \mu\right) / \sqrt{\hat{w}^T \mathbf{Q} \hat{w}} + \lambda (C.\mathbb{1} - \hat{w})^T . \hat{w}, \qquad (5)$$

where $\lambda > 0$ and $C > 1$ are two hyper parameters. One can find more details of how we can derive the loss function $L(\hat{w})$ in the section Supplementary Material. After that, by implementing an appropriate deep neural network to estimate the optimal solution $\hat{w} = (\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_N)$ in Eq. (5), we can derive the final optimal vector w^{opt} in Eq. (3) by the following rules:

$$w_i^{opt} = 1, \text{ if } \hat{w}_i \geq \theta, w_i^{opt} = 0, \text{ if } \hat{w}_i < \theta, \forall i = 1, \ldots, N. \qquad (6)$$

In our experiments, we choose $\theta = 0.5$.

2.3 Our Proposed Models for the Portfolio Optimization

To estimate the output vector $\hat{w}$, we consider different deep learning approaches for solving the portfolio optimization problem based on the proposed loss function in Eq. (5). We select both Long-short term memory(LSTM) [15] and Gate

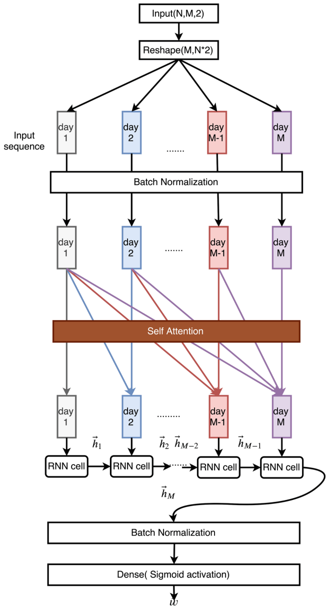


Fig. 2. Our proposed Self Attention + LSTM/GRU.

Recurrent Unit [16] architectures as two baseline models. Especially, by converting the input data into an $N \times M \times 2$ tensor as an "image", we construct a new ResNet architecture for the problem and create four other combinations of deep neural networks. They are SA + LSTM (Self-Attention model and LSTM), SA + GRU (Self-Attention model and GRU), AA + LSTM (Additional Attention and LSTM), and AA + GRU (Additional Attention and GRU). The architecture of RNN, GRU, and LSTM cells can be found more details at [15,16,21].

ResNet. ResNet architecture has been proven to become one of the most efficient deep learning models in computer vision, whose the first version was proposed by He et al. [22]. After that, these authors later released the second update for ResNet [14]. By using residual blocks inside its architecture, ResNet can help us to overcome the gradient vanishing problem and then well learn deep features without using too many parameters. In this work, we apply ResNet for estimating the optimal value for the vector $\hat{w}$ in the loss function (5). To the best of our knowledge, this is the first time ResNet is used for the Sharpe ratio maximization and our proposed ResNet architecture can be described in Fig. 4.

SA/AA + LSTM/GRU. The attention mechanism is currently one of the state-of-the-art algorithms, which are ubiquitously used in many NLP problems. There are many types of attention models, including Bahdanau attention [18], Luong Attention [21], and Self Attention [17]. Although the attention mechanism has been applying for the stock price prediction [23], there is few attention scheme used for maximizing the Sharpe ratio in the portfolio optimization problem. In this paper, we exploit two mechanisms, which are Self-Attention and Bahdanau attention (Additive Attention). The corresponding architecture of our four proposed models (SA + LSTM, SA + GRU, AA + LSTM, AA + GRU) can be visualized in Figs. 2 and 3.

3 Experiments

In this section, we present our experiments and the corresponding implementation of each proposed model. All tests are performed on a computer with Intel(R) Core(TM) i9-7900X CPU, running at 3.6 GHz with 128 GB of RAM, and two GPUs RTX-2080Ti (2 × 12 GB of RAM). We collect all stock data from the VN-HOSE stock exchange over six years (from January 1, 2013, to July 31, 2019) for measuring the performance of different models. There are 438 tickers appearing in the Vietnam stock market during this period. However, 57 tickers disappeared in the stock market at the end of 31/07/2019. For this reason, we only consider 381 remaining tickers for training and testing models. In Fig. 1(a) and 1(b), we visualize the mean values of both volume and price of the top 20 highest volume tickers in HOSE as well as the corresponding average value and the standard deviation of the daily return.

3.1 Model Configuration

In our experiments, all proposed models use the Adam optimizer [24] with the optimal learning rate $\alpha = 0.0762$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The learning rate

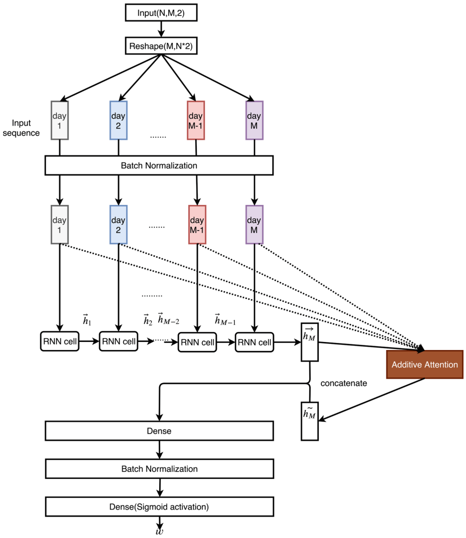


Fig. 3. Our proposed Additive Attention + LSTM/GRU.

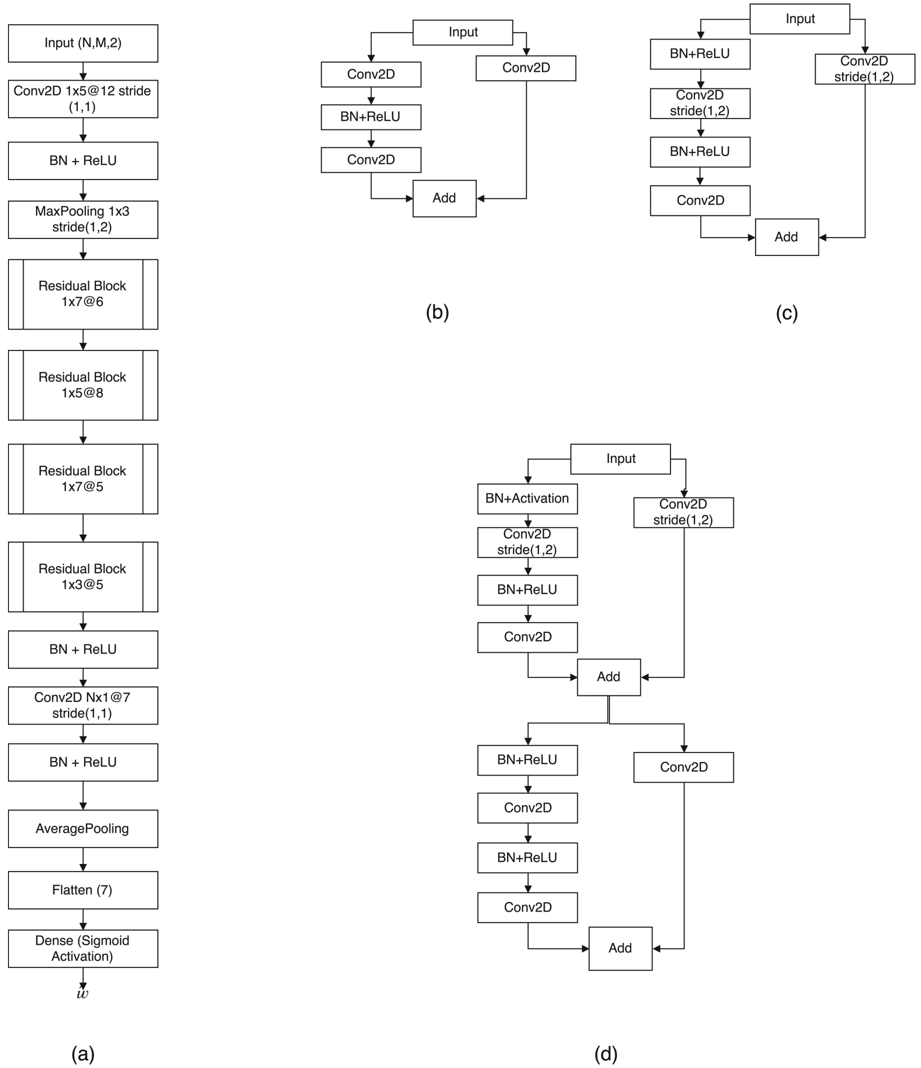


Fig. 4. (a) Our proposed Resnet model for the portfolio optimization problem. (b) The first residual block. (c) The second and the third residual block (d) The final residual block. Here, "BN" denotes "Batch Normalization", N is the number of tickers, and M is the number of days to extract the input data. In our experiments, $N = 381$ and $M = 64$.

and L2 regularization are tuned by using 141 random samples from the training set. We use the library Hyperas[2] for automatically tuning all hyper-parameters of the proposed models.

[2] https://github.com/maxpumperla/hyperas.

For two base line models (LSTM and GRU), we use 32 hidden units in which the L_2-regularization term is 0.0473. As shown in Fig. 4, our proposed ResNet model has the input data of the size $(381, 64, 2)$ passing to the first convolution layer where the kernel size is (1×5) and the L_2-regularization is 0.0932. After that, the data continue going through four different residual blocks, whose corresponding kernel sizes are (1×7),(1×5), (1×7), and (1×3), respectively, and all kernels have the L_2-regularization as 10^{-4}. Using these kernels, we aim at capturing the time dependency from the input data. The last convolution layer in our ResNet model has the kernel size $(381, 1)$ and the L_2-regularization as 0.0372 for estimating the correlation among all tickers. Its output data continue going through an average pooling layer before passing the final fully connected layer with the Sigmoid activation function to compute the vector $\hat{w}$. The last Dense layer has L2 regularization 0.099 and the learning rate of our ResNet model is 0.0256.

For four proposed models (SA/AA + LSTM/GRU), both Self-Attention and Additive Attention have 32 hidden units and the L_2-regularization term is 0.01. Both GRU and LSTM cells use 32 hidden units, the Sigmoid activation function, and the L_2-regularization as 0.0473. Two last fully connected layers have 32 hidden unites and the corresponding L_2-regularization is 0.0727. In our experiments, we choose $\theta = 0.5$, $\lambda = 0.003$, and $C = 1.6$, where θ, λ, and C are hyper-parameters of our proposed loss function.

3.2 Data Preparation

As there are only 381 tickers ($N = 381$) in the market at the end of the month July, 2019, we use the time windows of M consecutive days for extracting the input data of proposed models. On each day, we collect the information of both "price" and "volume" of these 381 tickers and $\mathbf{y}$, the daily return on the market in the next K days ($K = 19$). Consequently, the input data has the shape $(381, 64, 2)$ and we move the time window during the studying period of time (from January 1, 2013, to July 31, 2019) to obtain 1415 samples.

To deal with new tickers appeared, we fill all missing values by 0. For these missing data, our model may not learn anything from these data. Meanwhile, for the daily return in the next K days, we fill all missing values by -100. That is, as those tickers have been not disappeared yet, we set its daily return as a negative number so as to ensure chosen portfolios containing these tickers can get a negative Sharpe ratio. During training proposed models, we believe that the optimizer can learn well and avoid selecting these tickers from portfolios as much as possible.

3.3 Experimental Results

We evaluate each model by using 10-fold cross-validation or forward chaining validation in time series data. As shown in Fig. 5, while measuring the performance of each proposed model, we create the training data by moving the selected time window (64 days) during the investigating period (from January 2013 to July

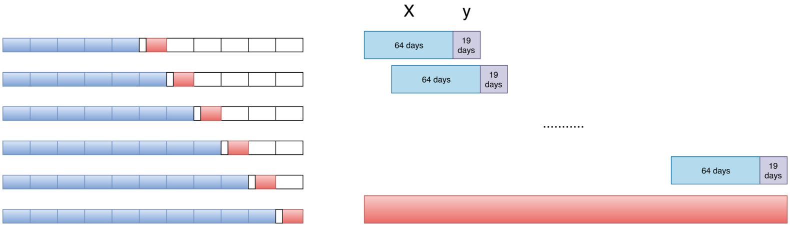


Fig. 5. The 10- Fold cross validation in our experiments. The blue blocks contain the training data and the red blocks contain the testing data. In experiments, we just use from Fold 5 to Fold 10 for evaluating the Sharpe ratio due to the lack of data for training models. At each fold, we train our deep models using 200 epochs. (Color figure online)

2019) and consider the corresponding sequence of daily returns in the next coming 19 days. It is crucial to make sure all training samples are independent of the testing samples.

The experiment results show that the Additive Attention + GRU model outperforms with the others. One of the possible reasons is Additive Attention + GRU can retain the information from the input sequence, which may loose from RNN cells when dealing with a very long sequence. Models using Self Attention can also get good results; however, as the outputs of the Self-Attention module still go to the RNN cell, without keeping any information from the sequence input. For this reason, the mean value of the Sharpe ratio of SA + GRU (0.9047) is a bit lower than AA + GRU (1.1056). Interestingly, the mean value of the Sharpe ratio of SA + LSTM (1.0206) is better than AA + LSTM (0.9235). Although not getting a high Sharpe ration in comparison with SA/AA + LSTM/GRU, the ResNet model has a quite short training time. In our experiments, the total time for its running 10-Fold Cross Validation is only 40 min, while taking over two hours for all SA/AA + LSTM/GRU models. In Fig. 6, our best-proposed model, AA + GRU, has a better performance than both VN30[3] and VNINDEX[4] in terms of the Sharpe ratio values. These experimental results show that our proposed techniques can achieve promising results and possibly apply not only in the Vietnamese stock market but also in other countries (Table 1).

[3] VN30 is the bucket of 30 companies having highest market capitalization and highest volume in six months for all the companies listed on the Ho Chi Minh City Stock Exchange. They also have the free float larger than 5%: https://iboard.ssi.com.vn/bang-gia/vn30.

[4] VN-Index is a capitalization-weighted index of all the companies listed on the Ho Chi Minh City Stock Exchange: https://www.bloomberg.com/quote/VNINDEX:IND.

Table 1. The performance of different models by the Sharpe ratio

Model	Number of learning parameters	mean (Sharpe ratio)	std (Sharpe ratio)
Resnet	66,967	0.8309	0.3391
LSTM	114,333	0,77057	0.2972
GRU	88,893	0.750	0.3182
SA + LSTM	164,689	1.0206	0.2976
SA + GRU	139,249	0.9047	0.3574
AA + LSTM	166,865	0.9235	0.2718
AA + GRU	141,425	**1.1056**	**0.2188**

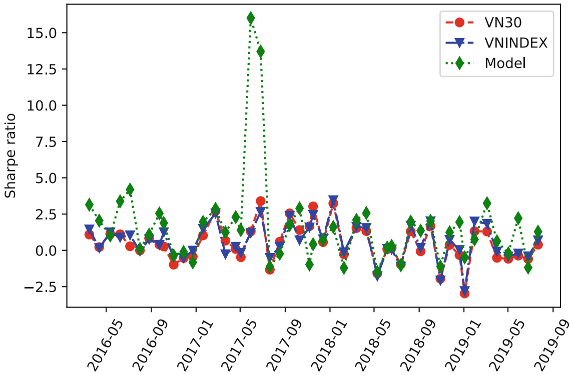


Fig. 6. The performance of our AA + GRU model with the VN30 and VNINDEX in terms of the Sharpe ratio in the testing dataset.

4 Conclusion and Further Work

We have proposed a novel approach for a portfolio optimization problem with N tickers by using the historical stock data during the last M days to compute an optimal portfolio that maximizes the Sharpe ratio of the daily returns during the next K days. We have also presented a new loss function for the Sharpe ratio maximization problem and transform the input data into a $N \times M \times 2$ tensor, and apply seven different deep learning methods (LSTM, GRU, SA + GRU, SA + LSTM, AA + LSTM, AA + GRU, and ResNet) for investigating the problem. To learning a suitable deep learning model for the problem, we collect the stock data in VN-HOSE during the period from January 2013 to July 2019. The experimental results show that the AA + GRU model outperforms with the other techniques and also achieves a better performance in terms of the Sharpe ratio for two popular indexes VN30 and VNINDEX.

In future works, we will extend our approaches to similar problems in other countries and continue improving our algorithms. Our project, including datasets and implementation details, will be publicly available[5].

Acknowledgement. We would like to thank The National Foundation for Science and Technology Development (NAFOSTED), University of Science, Inspectorio Research Lab, and AISIA Research Lab for supporting us throughout this paper.

References

1. Fernández, A., Gómez, S.: Portfolio selection using neural networks. Comput. Oper. Res. **34**(4), 1177–1191 (2007)
2. Chong, E., Han, C., Park, F.C.: Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. Expert Syst. Appl. **83**, 187–205 (2017)
3. Box, G.E.P., Jenkins, G.: Time Series Analysis, Forecasting and Control. Holden-Day Inc, San Francisco (1990)
4. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica **50**(4), 987–1007 (1982)
5. Mills, T.C.: Time Series Techniques for Economists. Cambridge University Press, Cambridge (1990)
6. Nguyen, B.T., Nguyen, D.M., Ho, L.S.T., Dinh, V.: An active learning framework for set inversion. Knowl. Based Syst. **185**, 104917 (2019)
7. Nguyen, D.M.H., Vu, H.T., Ung, H.Q., Nguyen, B.T.: 3D-brain segmentation using deep neural network and gaussian mixture model. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 815–824, March 2017
8. Cao, H.K., Ly, D.T., Nguyen, D.M., Nguyen, B.T.: Automatically generate hymns using variational attention models. In: Lu, H., Tang, H., Wang, Z. (eds.) ISNN 2019, Part II. LNCS, vol. 11555, pp. 317–327. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22808-8_32
9. Nguyen, B.T., Dang-Nguyen, D.-T., Dang, T. X., Phat, T., Gurrin, C.: A deep learning based food recognition system for lifelog images. In: Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods - Volume 1: INDEED, pp. 657–664. INSTICC, SciTePress (2018)
10. Liu, Q., Dang, C., Huang, T.: A one-layer recurrent neural network for real-time portfolio optimization with probability criterion. IEEE Trans. Cybern. **43**(1), 14–23 (2012)
11. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
12. Liu, J., Chao, F., Lin, Y.-C., Lin, C.-M.: Stock prices prediction using deep learning models, arXiv preprint arXiv:1909.12227 (2019)
13. Sharpe, W.F.: The sharpe ratio. J. Portfolio Manag. **21**(1), 49–58 (1994)
14. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part IV. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38

[5] https://github.com/caokyhieu/DELAFO-DeEp-Learning-Approach-for-portFolio-Optimization.

15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
16. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
18. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014)
19. Markowitz, H.: Portfolio selection. J. Finan. **7**(1), 77–91 (1952)
20. Kopman, L., Liu, S.: Maximizing the sharpe ratio, June 2009. In: MSCI Barra Research Paper, no. 2009–22 (2009)
21. Luong, M.-T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025 (2015)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
23. Li, H., Shen, Y., Zhu, Y.: Stock price prediction using attention-based multi-input LSTM. In: Asian Conference on Machine Learning, pp. 454–469 (2018)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014)