Proceedings of Science and Mathematics
Faculty of science, universiti teknologi malaysia

https://science.utm.my/procscimath
Volume 21 (2024) 31-39

FORECASTING STOCK MARKET PRICES USING Q-LEARNING MODELS

Chen Jian Kai, Norma Alias*
Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia
*Corresponding author: normaalias@utm.my

Abstract

Predicting the stock market involves attempting to forecast the future worth of a company's stock or any other financial asset traded on an exchange. Accurately predicting a stock's future price could result in substantial financial gain. According to the efficient-market hypothesis, stock prices incorporate all presently accessible information, implying that any alterations in price not grounded in newly disclosed information are essentially unforeseeable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information. With the rise of digital computing, the field of stock market prediction has transitioned into the realm of technology. Generally, the main agenda of this research is to utilize currently state-of-the-art deep learning models that are currently available in the market and predict the prices of the stocks each day. We use reinforcement learning model by applying Q-Learning to predict whether the stock is worth buying or selling in certain periods based on different investment strategies. To do this, we choose several stocks available in Bursa Malaysia, then predict their price from 2010 to 2017. The performance is assessed via different perspectives: overall profitability, maximum profitability and time to profit. Then, we compare the results with supervised machine learning models such as random forest. Based on the results, we can see that the models are very accurate and hold well in real market simulations, as well as do transactions accordingly.

Keywords: Stock market; Forecasting; Returns; Machine learning; Artificial Intelligence; Artificial Neural Network; Reinforcement learning; Q-Learning; Supervised learning; Random Forest; Long short-term memory

Introduction

A stock exchange serves as a marketplace where traders and stockbrokers engage in buying and selling various financial instruments like equity stocks, bonds, and securities. Having a company's stocks listed on such an exchange enhances their liquidity, making them more appealing to a wider array of investors. Additionally, the exchange often plays a role in ensuring secure settlement transactions. Some companies opt to list their stocks on multiple exchanges across different countries to attract international investors.

Stock exchanges can encompass diverse securities beyond stocks, including bonds and occasionally derivatives, which tend to be traded through dealers outside the formal exchange setting.

Trading in stock markets involves the exchange of stocks or securities between a seller and a buyer, with both parties agreeing on a price. Stocks, or shares, represent ownership in a specific company.

The stock market accommodates a spectrum of participants, ranging from individual investors to large institutions situated globally, including banks, insurance companies, pension funds, and hedge funds. These participants may authorize a stock exchange trader to execute their buy or sell orders on their behalf.

Predicting the stock market involves attempting to forecast the future worth of a company's stock or any other financial asset traded on an exchange. Accurately predicting a stock's future price could result in substantial financial gain. According to the efficient-market hypothesis, stock prices incorporate all presently accessible information, implying that any alterations in price not grounded in newly disclosed information are essentially unforeseeable. Others disagree and those with this viewpoint possess myriad methods and technologies which purportedly allow them to gain future price information.

With the rise of digital computing, the field of stock market prediction has transitioned into the realm of technology. The leading method involves employing artificial neural networks (ANNs) and genetic algorithms (GAs). Research indicates that the bacterial chemotaxis optimization method might outperform GA. ANNs can be likened to mathematical approximators, with the prevalent type being the feedforward network utilizing the backward propagation of errors algorithm to adjust network weights—often termed backpropagation networks. Another suitable ANN type for stock prediction is the time recurrent neural network (RNN) or time delay neural network (TDNN), examples being the Elman, Jordan, and Elman-Jordan networks.

Regarding ANNs in stock prediction, two common approaches exist for forecasting different time spans: independent and joint. The independent approach employs a separate ANN for each time span (e.g., 1-day, 2-day, or 5-day), minimizing the impact of forecasting error on other spans as they are distinct problems. Conversely, the joint approach synchronously determines multiple time spans, potentially leading to error sharing between spans and a higher risk of overfitting due to increased parameters.

Recent academic research on ANNs for stock forecasting predominantly favors an ensemble of independent ANN methods, showing more success. This ensemble might utilize different networks—one predicting future lows using low prices and time lags, while another forecasts highs using lagged highs. These predicted lows and highs then establish stop prices for buying or selling. Additionally, outputs from individual "low" and "high" networks could feed into a final network that incorporates volume, intermarket data, or statistical price summaries, culminating in an ensemble output triggering buying, selling, or directional market changes. An essential discovery with ANNs and stock prediction is that employing a classification approach (such as buy (+1) and sell (-1)) tends to yield greater predictive reliability compared to quantitative outputs like low or high prices.

Generally, the main agenda of this research is to utilize currently state-of-the-art deep learning models that are currently available in the market and predict the prices of the stocks each day.

Literature Review

Artificial Neural Network (ANN)

Artificial neural networks (ANNs), also known as neural networks (NNs) or neural nets, form a subset of machine learning models inspired by the organizational principles found in biological neural networks present in animal brains.

These ANNs comprise interconnected units known as artificial neurons, roughly resembling biological brain neurons. Similar to synapses transmitting signals in a biological brain, each connection in an artificial neuron network can convey signals to other neurons. An artificial neuron receives, processes, and can transmit signals to connected neurons. The transmitted "signal" is represented as a real number, and each neuron's output results from a non-linear function applied to the sum of its inputs. These connections are termed edges, with neurons and edges often possessing weights that adjust during the learning process, influencing the signal's strength. Neurons might include a threshold, allowing signal transmission only if the combined signal surpasses that threshold.

Typically, neurons are organized into layers, with each layer potentially executing distinct transformations on its inputs. Signals progress from the initial layer (the input layer) to the concluding layer (the output layer), possibly traversing through multiple layers during this journey.

Q-Learning

Q-Learning Model is a type of reinforcement learning algorithm that does not need a specific model to learn the value of an action in each state. No model is used and it can be able to handle problems with stochastic transitions and rewards without requiring adaptations. This model will look for best policy to maximize the total reward value and able to identify an optimal action-selection policy for any given finite Markov decision process, while taking infinite time to explore with certain randomness.

To initialize the Q-values, the model initializes $Q(s,a)$ for all state-action pairs to some initial value, typically equals to zero. For the current state *s*, it will choose an action *a* based on a policy derived from the current Q-value, such as ε-greedy policy. Then, it will perform the action *a* and let the environment return the reward *r* and the new state *s'*. The Q-value is then updated for the state-action pair using the following update rule:

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

where:

- *α* is the learning rate
- *γ* is the discount factor.
- *r* is the reward received when moving from the state *s* to the state *s'*.
- $\max_{a'} Q(s',a')$is the maximum Q-value over all possible actions *a'* in the new state *s'*

The model repeats all over again until the Q-values converge, or a certain number of episodes have been completed.

Supervised Learning

Supervised learning uses input objects and desired output value to train a model. The model will process the data and maps new data on expected output by creating suitable function. A well thought out situation will aid the model to determine the correct outcome for unseen instances, which needs the algorithm to generalize from the training data to unseen instances in best possible attempt.

The training of the model typically trickles down to several steps. The user needs to find the best data to let the model to train on, this kind of dataset must be following the real-world use of the function. This means we need to collect the input objects and corresponding output values, by using scientific measurements or human interpretations. Next, the user needs to determine the input feature representation of the learned function, which must have sufficient information to predict accurate output. The structure of the learned function and corresponding learning algorithm must be decided as well. Once the design is done, we execute the algorithm on the training data and its accuracy will be evaluated with post adjustment on parameters.

Random Forest

Random forest or random decision forest is one of the supervised ensembles learning methods for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. [Ho, 1995] This technique helps fixing the overfitting issues that occurred on decision trees. [Hastie, Tibshirani, Friedman, 2008]

The algorithm will use bootstrap aggregating, or also known as bagging, by using a given training set *X* = x_1, ..., x_n with responses *X* = x_1, ..., x_n, which will be bagged repeatedly (*B* times) to choose a random sample with replacement of the training set and fit the tree to these samples. Each decision tree is built on a different sample of data. These samples are created by random sampling with replacement, also known as bootstrapping. At each node of a decision tree, a random subset of features is selected to determine the best split. Via prediction, the model will collect the results from all decision trees. This process will provide accurate and stable results thanks to the collaborative decision-making process, with the insights of multiple trees.

Mathematically, if we denote the output of the m-th tree as $h_m(x)$, where *x* is the input vector, then the final output of the Random Forest for a classification problem is given by:

$$H(x) = \text{majority}\{h_1(x), h_2(x), \ldots, h_M(x)\}$$

And for a regression problem, it is given by:

$$H(x) = \frac{1}{M}\sum_{m=1}^{M} h_m(x)$$

where M is the total number of trees.

Methodology

Research Activities

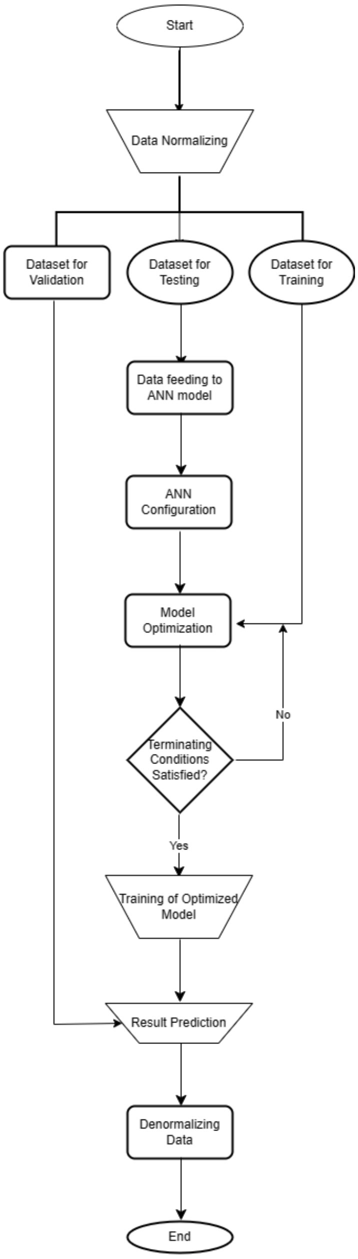
The historical data used will be Maybank (1155) in Bursa Malaysia. The time interval will be from 1/1/2010 to 31/12/2017. It will be used to train the data by iterations or episodes to find the best way to obtain profit throughout the period. We will also be creating input variables and output variables for training the model.

To design the model, we will choose the learning rate, discount factor and the initial conditions. The model is then trained using the training dataset and tune hyperparameters, while experimenting with different architectures, learning rates, batch sizes and etcetera. We will also perform cross-validation to prevent overfitting and fine-tune the model.

The model's performance is then evaluated via several criteria, which is maximum profit for each episode and average profit per episode. Afterwards, the model will be compared with one supervised deep learning model which is Random Forest.

Based on the results from evaluation and testing, the model will be refined and experimented with different feature sets and hyperparameters.

Operational Flow Chart depicting the Research Structure of this Study



Results and discussion

Table 4.1: Maximum Profit Earned in Selected Episodes

Episode	Maximum Profit Earned (RM)	Profit Percentage (%)	Maximum Profit Achieved Date	Trading Days until Maximum Profit
1	13023.97	30.23	15/10/2012	687
50	10996.37	9.96	11/10/2010	193
100	13875.39	38.75	4/6/2013	841

Based on the table above, we can see that we managed to obtain profit, albeit in different margin and in different period. This is mainly because the model is preset to explore all possible strategies, which means that the algorithm will not improve over time. Instead, the best outcome will be lying in one of the 100 episodes.

Based on the result data, we find that episode no. 19 has the best strategy, with over RM 20,683.94 maximum earned over a period of 1868 days or on 26/7/2017. This nets a profit of over 106.83% of profit, which is very impressive considering the average profitability for the episodes are around 30%, although it takes longer time to achieve such result.

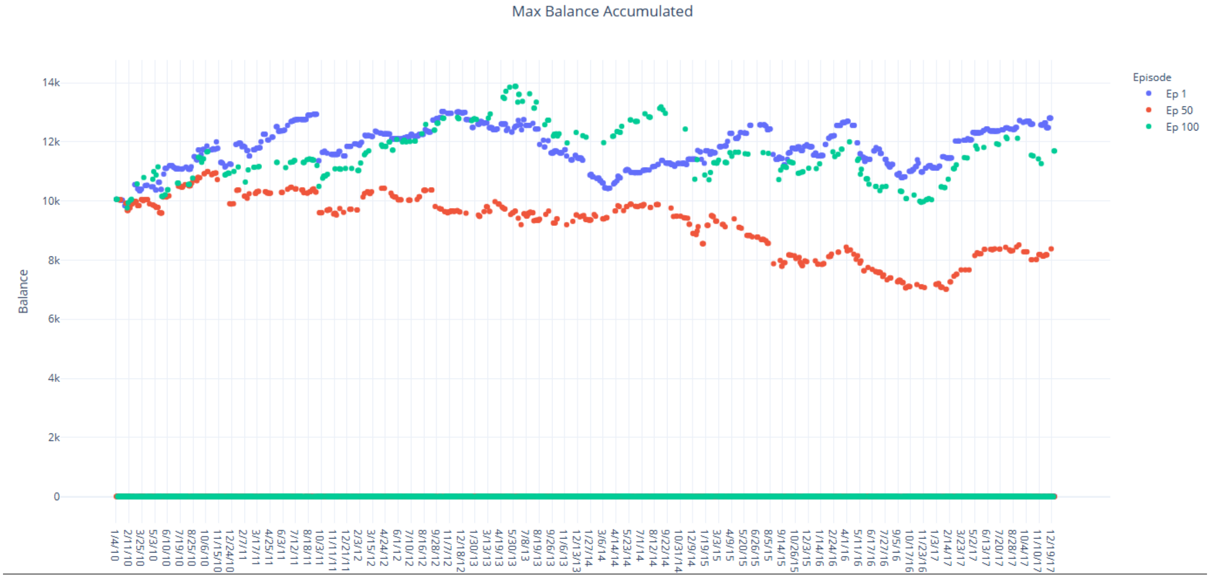


Figure 4.1 Maximum Profit Gained Over the Period

Based on the graph, we can see that while the selected three episodes are able to gain balance more than initially provided value, we can see that there will be fluctuations over the time. For example, on episode no. 50, we can see that the trend of the balance is going downward, and even leads to negative profit. This means the strategy applied on this episode is viable on the long run. On the other hand, both episode no. 1 and no. 100 have an upward trend in terms of daily balance, which means the strategy worked in our favour.

On the next graph, we compare the best episode (Episode no. 19) which has the most maximum profit gained in one day with episode no. 100 to compare the best episode that can be used throughout the training period.

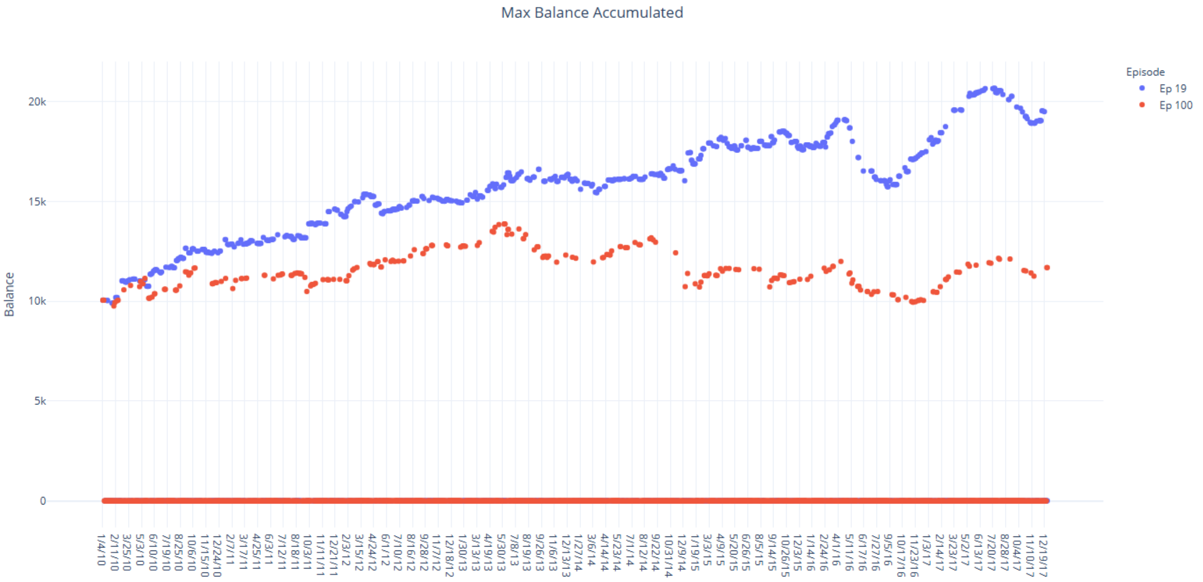


Figure 4.2 Episode no. 19 vs Episode no. 100

We can see that episode no. 19 not only beats episode no. 100 in terms of maximum profit gained, but it also beats long term profitability as well, albeit with slight decreasing around second half of 2016.

Based primarily on the analysis above, we can tell that as a type of reinforcement learning, Q-Learning model is very well suited to perform stock market trading, with its performance overwhelmingly won over traditional analysis by trading experts and professionals. Additionally, we can use the best performing episode to retrain the same model in hope of finding the better strategy that can be put to the test. While we only look for 4 out of the 100 episodes for comparison and reference, we are certainly sure that there might be an even better episode that can overperform episode no. 19.

Next, we compare Q-Learning model with Random Forest model. We choose the best performing episode from each model for reference.

Table 4.2: Comparing Best Episodes between Q-Learning model and Random Forest model

Model	Maximum Profit Earned (RM)	Profit Percentage (%)	Maximum Profit Achieved Date	Trading Days until Maximum Profit
Q-Learning	20683.94	106.83	26/7/2017	1868
Random Forest	17273.11	72.73	18/7/2013	873

Based on the table, we can see that Q-Learning model is leading in terms of maximum profitability and least time to achieve maximum profitability. Both models are able to gain positive profit but in a different period time, Q-Learning model takes longer which is near the end of training period. This means that both episodes have viable strategies for long term investors.

The next graph **will be comparing their profitability throughout the period.**

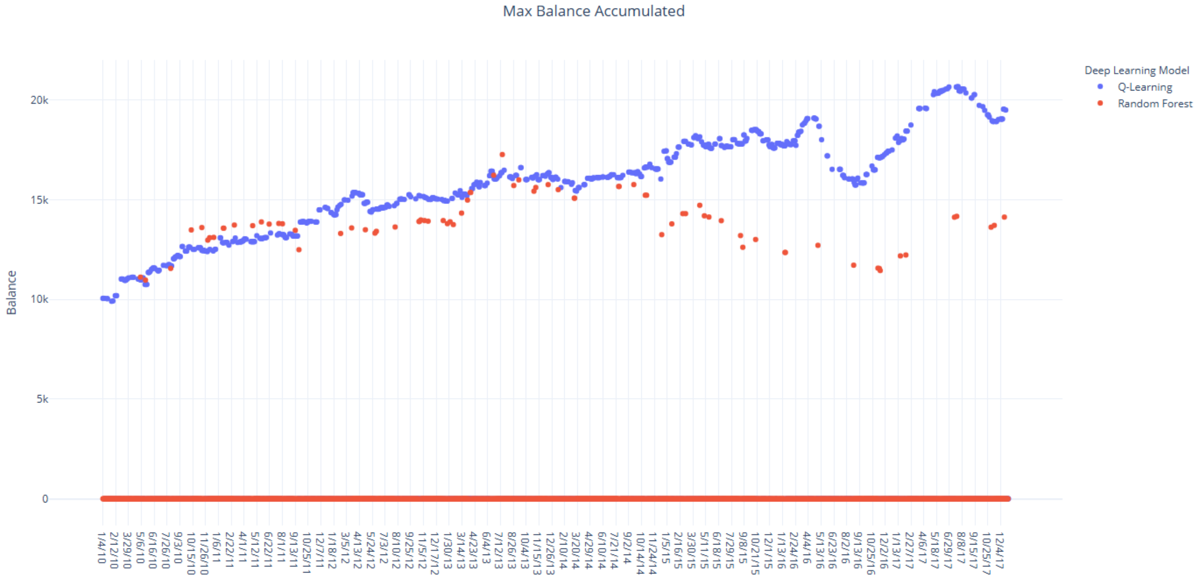


Figure 4.3 Comparison between Both Models for Profit over Period

Based on the graph, we can see that Q-Learning still leading the place in terms of over the period profitability. We can also see that Q-Learning model is more aggressive in doing transactions based on the frequent drops of the blue line, that indicates frequent changes of the cash balance on hand. Whereas Random Forest model tend to hold the stocks until the right time to sell based on the infrequent drop of the red line. This means that Random Forest model mimics more to human behaviour where investors tend to hold the stocks until they deemed the right time to sell off, and Q-Learning model will be more robot-like which tends to be more proactive to gain more profit as possible

Conclusion

This research has proved that artificial intelligence, or in particular, deep learning models are indeed a viable approach to forecast stock market prices and predict the next move in order to perform buy low, sell high strategy. The Q-Learning model we developed is indeed usable and performs well in our needs. It can be used from short to medium to long term investment while respecting the investor's risk appetite should it be applicable.

Acknowledgement

I would like to express my biggest gratitude to all the people that has provided me with strength, encouragement, and guidance that are much needed throughout the research. Special thanks to my supervisor, Dr Norma binti Alias, my friends and family, and UTM for all the constant support.

References

[1] Avramelou, L., Nousi, P., Passalis, N., & Tefas, A. (2024). Deep reinforcement learning for financial trading using multi-modal features. Expert Systems With Applications, 238, 121849. https://doi.org/10.1016/j.eswa.2023.121849

[2] Bas, E., Egrioglu, E. & Kolemen, E. Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization. Granul. Comput. 7, 411–420 (2022). https://doi-org.ezproxy.utm.my/10.1007/s41066-021-00274-2

[3] Bukhari, A. H., Raja, M. A. Z., Sulaiman, M., Islam, S., Shoaib, M., & Kumam, P. (2020). Fractional Neuro-Sequential ARFIMA-LSTM for Financial Market Forecasting. IEEE Access, vol. 8, pp. 71326-71338. doi: 10.1109/ACCESS.2020.2985763.

[4] Cheng, D., Yang, F., Xiang, S., & Liu, J. (2022). Financial time series forecasting with multi-modality graph neural network. Pattern Recognition, 121, 108218. https://doi.org/10.1016/j.patcog.2021.108218

[5] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). *The Elements of Statistical Learning* (2nd ed.). Springer. ISBN 0-387-95284-5.

[6] Ho, Tin Kam (1995). *Random Decision Forests* (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

[7] Jiang, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Systems With Applications, 178, 115019. https://doi.org/10.1016/j.eswa.2021.115019

[8] Li, Shengbo (2023). *Reinforcement Learning for Sequential Decision and Optimal Control* (First ed.). Springer Verlag, Singapore. pp. 1–460. doi:10.1007/978-981-19-7784-8. ISBN 978-9-811-97783-1. S2CID 257928563.

[9] Li, Y., Pan, Y. (2022). A novel ensemble deep learning model for stock prediction based on stock prices and news. Int J Data Sci Anal 13, 139–149. https://doi-org.ezproxy.utm.my/10.1007/s41060-021-00279-9

[10] Majidi, N., Shamsi, M., & Marvasti, F. (2024). Algorithmic trading using continuous action space deep reinforcement learning. Expert Systems With Applications, 235, 121245. https://doi.org/10.1016/j.eswa.2023.121245

[11] Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D., & De, S. (2021). Stock market prediction using deep learning algorithms. CAAI Transactions on Intelligence Technology, 8(1), 82–94. https://doi.org/10.1049/cit2.12059

[12] Oyedele, A. A., Ajayi, A. O., Oyedelec, L. O., Bello, S. A., & Jimoh, K. O. (2023). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. Expert Systems With Applications, 213, 119233. https://doi.org/10.1016/j.eswa.2022.119233

[13] Pang, X., Zhou, Y., Wang, P. et al. (2020). An innovative neural network approach for stock market prediction. J Supercomput 76, 2098–2118. https://doi-org.ezproxy.utm.my/10.1007/s11227-017-2228-y

[14] Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. Expert Systems With Applications, 169, 114332. https://doi.org/10.1016/j.eswa.2020.114332

[15] Tao, Z., Wu, W., & Wang, J. (2024). Series decomposition Transformer with period-correlation for stock market index prediction. Expert Systems With Applications, 237, 121424. https://doi.org/10.1016/j.eswa.2023.121424

[16] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science, 167, 599–606. https://doi.org/10.1016/j.procs.2020.03.326

[17] Zaheer, S., Anjum, N., Hussain, S., Algarni, A. D., Iqbal, J., Bourouis, S., & Ullah, S. S. (2023). A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model. Mathematics, 11(3), 590. MDPI AG. Retrieved from http://dx.doi.org/10.3390/math11030590

[18] Zhang, D., & Lou, S. (2021). The application research of neural network and BP algorithm in stock price pattern classification and prediction. Future Generation Computer Systems, 115, 872–879. https://doi.org/10.1016/j.future.2020.10.009