MJFAS
Malaysian Journal of Fundamental and Applied Sciences

RESEARCH ARTICLE

Improving Stock Price Forecasting Accuracy with Stochastic Multilayer Perceptron

Assunta Malar Patrick Vincent, Hassilah Salleh*

Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia

Abstract The stock market operates in a stochastic environment, making accurate price forecasting challenging. To address this issue, a stochastic multilayer perceptron (S-MLP) model has been developed to simulate the stock market's stochastic nature. By incorporating a Gaussian process into the sigmoid activation function, this model incorporates stochasticity into the traditional multilayer perceptron (MLP). As the perturbation factor, a stochastic sigmoid activation function (SAF) with a volatility estimator is used. Although S-MLP has demonstrated superiority over MLP, there is still room for improvement in terms of forecasting precision. In this study, we propose S-MLP with a trainable perturbation factor (S-MLPT), an improved variant of S-MLP. SAF employs the Yang-Zhang volatility estimator as the perturbation factor. The proposed model first employed MLP, and all the parameters were trained. After freezing the parameters, S-MLP is used to train the perturbation factor in the SAF. To evaluate the predictive performance of the models, MLP, S-MLP, and S-MLPT are used to predict the one day ahead highest stock price of four counters listed in Bursa Malaysia. As an evaluation metric, the coefficient of determination is utilised, and the relative percentage improvement of the models is calculated to determine their superiority. The results demonstrated that S-MLP outperforms MLP by effectively minimizing the loss function and converging towards a better local or global minimum during training. In conclusion, S-MLPT exhibits even better performance than S-MLP, with relative percentage improvements of 0.14%, 15.45%, and 0.48% for counters 0166.KL, 2445.KL, and 4707.KL, respectively.

Keywords: Forecasting stock price, deep learning, multilayer perceptron, stochastic multilayer perceptron.

Introduction

Stock markets are influenced by random political and economic events [1]. These events generate a noisy environment and uncertainty in the stock market. Stock prices are classified as one of the "noisiest" time series and the volatility in stock prices is caused by a stochastic process, that is noisy, dynamic, non-linear, non-parametric, non-stationary, and chaotic [2]. Hence, it is challenging to forecast the future price of the stock [2 - 3], yet it is not impossible.

As a result, numerous mathematical models have been created and applied on financial market forecasting. Statistical approaches are linear in nature, it hinders prediction performances in case of sudden rises or fall in stock prices [4], and they fail to capture the non-linear pattern present in the data [5 - 7]. These disadvantages were solved by using deep learning models into financial forecasting. A deep learning model can automatically extract features from noisy and dense data to find hidden non-linear relationships [8] and approximate all the internal parameters through incremental learning [9]. To overcome the shortcomings of the linear forecasting, deep neural networks are preferred to solve some of the problems posed by financial market forecasting with a statistical model.

Multilayer perceptron (MLP) model, widely regarded as highly significant, has undergone evolutionary development from the class of shallow neural networks known as ANN [10]. According to the authors,

***For correspondence:**
hassilah@umt.edu.my

Received: 27 March 2024
Accepted: 29 July 2024

MLP models have been utilized in forecasting research across various disciplines. In the financial market, extensive research has been applied on MLP to forecast stock prices. Among the latest research are those by [10-15]. From these literatures, it is worth noting that the stock market is highly complex and unpredictable, and no single model can accurately predict future stock prices with 100% certainty. Deep learning model has several benefits comparative to other forecasting models which are not deep learning. This is because it exhibits better adaptive capabilities, efficient training process, and nonstationary signal processing capabilities [16]. These characteristics makes deep learning models to be effective in solving non-linear problems that may pose challenges [17]. It has undergone significant advancement in its model development, particularly in the context of time series forecasting to address targeted problems.

One disadvantage of applying deep learning to forecast stock prices is the possibility that the loss function may not reach the global minimum or optimal local minimum. It can be encountered by introducing stochasticity into MLP during training process as the optimization algorithm will allows the loss function to escape poor local minima or the saddle point during the training phase of the neural network [18]. Among the MLP models that have hybridised stochasticity in the application of forecasting financial markets are Stochastic Time Effective Neural Network (STNN) [19]; the extension of STNN by [20-24]; and Stochastic Neural Network [25]. STNN and the extension of STNNs incorporated Brownian motion (BM) into the loss function of the neural network. Meanwhile, another study adapted random walk theory to the activation function of the neural network [25]. These studies emphasized that stochasticity was introduced into the deterministic neural network so it could mimic and adapt to the trend of the financial market without changing the original trend. In addition, the outcome of these studies concluded that stochastic models have better accuracy of forecasting in comparison to their deterministic counterparts. Therefore, incorporation of stochasticity into MLP not only allows the loss function to minimize well during training, but it also represents the stock market environment in the model.

Stochasticity was incorporated into the sigmoid activation function of MLP [26] which inspired from [18]. Stochastic sigmoid activation function was developed by integrating Gaussian process with the perturbation factor of volatility estimators (Roger-Satchell and Yang-Zhang) derived from respective stock prices. The SAF was then applied to a stochastic multilayer perceptron (S-MLP) to forecast the highest stock price one day ahead for eight counters listed in Bursa Malaysia. The results showed that the proposed network performed inferiorly compared to MLP, except for several counters. Hence, this study aims to further investigate the learning curve of the neural network used for forecasting the one day ahead highest stock price of the selected counters.

Hybridized stochastic model performed slightly poorer in comparison to the deterministic MLP and LSTM when random walk is integrated. To improve the performance of the model the stochastic parameter was trained via gradient descent backpropagation [25]. However, research by [26] has only incorporated stochasticity into MLP to forecast the stock prices. Considering this as a research gap this research aims to train the stochastic parameter in S-MLP to further improve the accuracy of the model.

In summary, this research aims to achieve two objectives. Firstly, it further investigates the learning curve of the S-MLP comparative to MLP employed for forecasting one day ahead highest price of the selected stock counters. Secondly, drawing inspiration form the identified research gap in the previous study [26], this research takes a step further by training the stochastic parameter in S-MLP. The remainder of the manuscript is structured with the description of the dataset, pre-processing steps and the employment of the predictive model in section 2. Results and discussions are presented in section 3. Finally, section 4 draws a conclusion based on the outcome of the research and recommendation of the future direction of the research.

Methodology

Data Collection

Various intervals of historical stock price data with a specific period can be collected to forecast it. Recent applications of forecasting stock price used different periods of data with a daily interval. For example, approximately three years of daily stock indices [27], three years of daily data of iShares MSCI United Kingdom [28] and five years of data [29]. In addition, the choice of dataset mainly depends on the choice of the researcher and usually contains hundreds or thousands of observations [30]. Thus, in this research, approximately five years of stock price and volume of traded stock from 6th March 2017 to 25th May 2022 is collected, which amounts to 1288 data. Daily historical prices and volume of stock traded consist of four counters: Inari Amertron Berhad (0166.KL), Kuala Lumpur Kepong Bhd (2445.KL), PPB Group Bhd (4065.KL) and Nestle (M) Bhd (4707.KL). The historical data consist of the date along with the opening and closing prices, the highest and lowest prices of the respective trading day, and the volume of the total traded shares. This study forecast one a day ahead highest price so, daily opening, closing, highest, lowest prices and volume of traded stock are the input features.

Data Pre-processing

Data pre-processing is the initial stage before employing the predictive model. First, to clean the collected data, the missing values were identified and dropped to avoid errors in the propagation during the training. Then, normalization of each feature over [0,1] ranges is done to remove the negative effects of data dispersed over various ranges [31]. Finally, the data is allocated to training, validation and testing set before loading them into deep learning model.

It is elusive to determine a perfect proportion for the amount of data required in training and testing sets [32]. However, the fundamental guideline is that the test and validation sets should be large enough to adequately represent the entire range of variability that exists in the datasets. Then, the remaining data can be utilized for training. Usually, the ratio of training set should be larger than validation and testing set as the number of datasets increases. Besides, the ratio of training, validation and testing should grow smaller. In the research by Sagir and Sathasivan [33] and Yassin *et al*. [34] allocated the training set with 68% and 70% ratio. Meanwhile, the remaining data was split equally to validation and testing set in both these studies. Taking in account of the past research, this study used 70% and 20% of data for training and validation respectively and remaining data is used to test the accuracy of the neural networks. The purpose of this stage is to convert the unprocessed data into a format that is simpler and more efficient for utilization in predictive modelling. The structural overview of data pre-processing is shown in Figure 1.

Predictive Models

This research considers three neural networks which are MLP (baseline model), S-MLP and the improvement of S-MLP; S-MLP with a trainable perturbation factor (S-MLPT). All the neural networks have the same architecture, which has five input nodes, one output node, and two hidden layers, with four and two nodes in the first and second hidden layers, respectively as shown in figure 2. The architecture of the neural network is decided by using the geometric pyramid rule proposed by Masters [35]. In addition, the value of weight is initialized using the Glorot normal initializer, which was proposed by Glorot & Bengio [36]. Besides that, all the neural networks were trained via the backpropagation algorithm and optimized with stochastic gradient descent. The training is an iterative process, where a small update will be done to the weight and bias parameters during each iteration. During the iterative process, the error between the output of the neural network and the target value is calculated using the mean square loss function. To stop the training of neural network, this study adopted early stopping mechanisms to prevent overfitting. The entire training phase is done using the data from training set. Finally, the data from the test set will be used to forecast the one day ahead highest price. To ensure consistency and eliminate biases in the accuracy and performance of the neural networks, this study maintains a constant set of parameters. The structural overview of MLP and S-MLP, and S-MLP^{T} is shown in figure 3 and 4, respectively.

The key differences between MLP and the variation of S-MLP differs based on the use of the activation function. During the training and testing of MLP, rectified linear unit (ReLu) activation function is used, as shown in Equation (1). Meanwhile, SAF shown in Equation (2) which is used in S-MLP.

$$\Phi_r(z) = \max(0, z) \tag{1}$$

$$\Phi_{SAF}(z, \xi) = \frac{1}{1 + e^{z}} + (\hat{\sigma}_{YZ})\xi \tag{2}$$

$$\hat{\sigma}^2{}_{YZ} = \hat{\sigma}^2{}_{o} + k\hat{\sigma}^2{}_{c} + (1 - k)\hat{\sigma}^2{}_{RS}\ , \qquad k = \frac{\alpha - 1}{\alpha + \frac{n+1}{n-1}} \tag{3}$$

Where, $\Phi_r(z)$ is ReLu activation function with z as the output of each hidden layer. Besides, $\Phi_{SAF}(z, \xi)$ is SAF with $\hat{\sigma}^2{}_{YZ}$ Yang-Zhang volatility estimator. k is a constant with the α=1.34 as suggested by Vințe & Ausloos [37] and Yang & Zhang [38]. n is the number of days in the sample period. $\hat{\sigma}^2{}_{o} = \frac{1}{n-1}\sum_{t=1}^{n}\left(o_t - \frac{1}{n}\sum_{t=1}^{n} o_t\right)^2$ and $\hat{\sigma}^2{}_{c} = \frac{1}{n-1}\sum_{t=1}^{n}\left(c_t - \frac{1}{n}\sum_{t=1}^{n} c_t\right)^2$ is standard volatility of opening and closing price of the stock counters, respectively. $\hat{\sigma}^2{}_{RS} = \frac{1}{n}\sum_{t=1}^{n} h_t(h_t - c_t) + l_t(l_t - c_t)$ is Rogers-Satchell volatility estimator. Furthermore, $o_t = \ln O_t - \ln C_{t-1}$, $c_t = \ln C_t - \ln O_t$, $h_t = \ln H_t - \ln O_t$, and $l_t = \ln L_t - \ln O_t$ are normalized opening, closing, of stock counter, respectively. Meanwhile, $O_t, C_t, H_t\ and\ L_t$ are the opening, closing highest and lowest price of the stock on date t.

The proposed S-MLP^{T} is adapted from the work of Jay *et al.* [25], where initially the weight and bias parameters were trained using a deterministic approach. Then, all parameters, except the perturbation factor, were frozen, and the value of the perturbation factor was optimized through error backpropagation with gradient descent. Following a similar rationale, this study trained MLP with ReLu activation function via backpropagation algorithm and optimized using SGD to obtain the updated values of weights and biases. Once the values of the weights and biases were frozen, the S-MLP was employed to train the perturbation factor, $\hat{\sigma}_{YZ}$, is trained through backpropagation with SGD optimizer. Using the updated value of the perturbation factor, the study then proceeded to forecast the one day ahead highest price of the stock counters.

Finally, the performance of the model is evaluated by calculating the coefficient of determination (R^2), shown in Equation (7). R^2 indicates how much of the target values are being captured by the model. R^2 usually shows the percentage variance that the independent variables account for in the dependent variable's variance. Moreover, R^2 is bounded to the range of $(-\infty, 1]$. It is stated that when the value of R^2 approaches the upper bound it means that the predictive model results in accurate forecasting and vice versa if R^2 approaching zero. However, if R^2 results in a negative value that it strongly suggested that the given predictive model has the worse fit [39]

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{\sum_{t=1}^{n}(\bar{y} - y_t)^2} \tag{7}$$

Where, $\hat{y}_t$ is the forecasted value, y_t is the corresponding actual value on day t. $\bar{y}$ is the mean on the data in test set.

To identify which neural network has better accuracy of one day ahead highest stock price forecasting a comparative analysis between MLP, S-MLP and S-MLP^{T} is carried out by calculating the relative improvement of the models using the value as shown in Equation (8).

$$\frac{R^2\ of\ the\ selected\ model - R^2\ of\ baseline\ model}{R^2\ of\ baseline\ model} \times 100 \tag{8}$$

Figure 1. Structural overview of data pre-processing

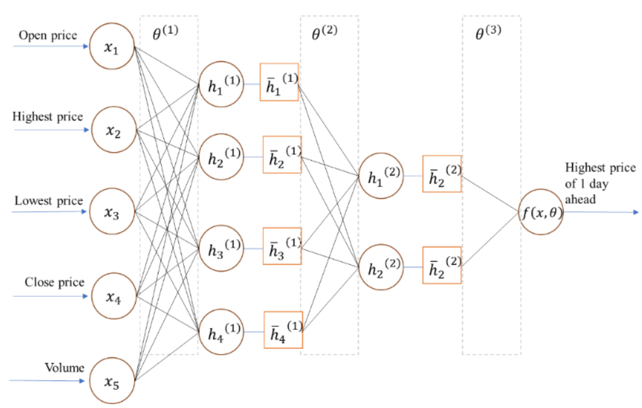

Figure 2. Architecture of the neural networks

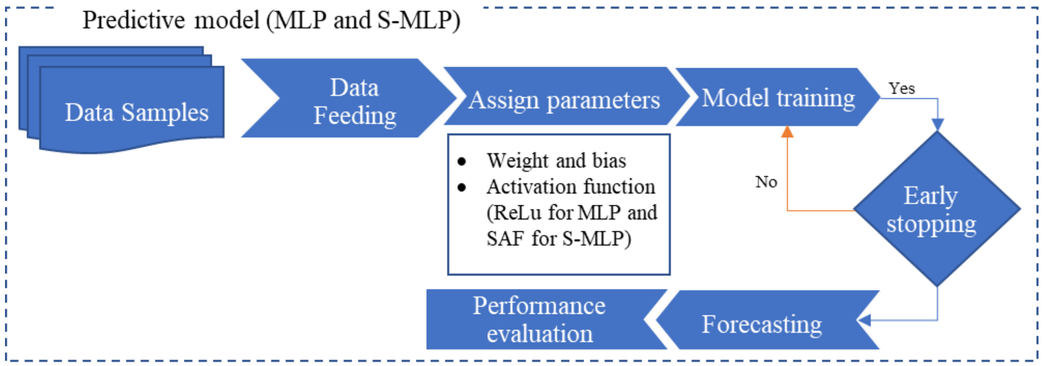

Figure 3. Structural overview of MLP and S-MLP predictive model

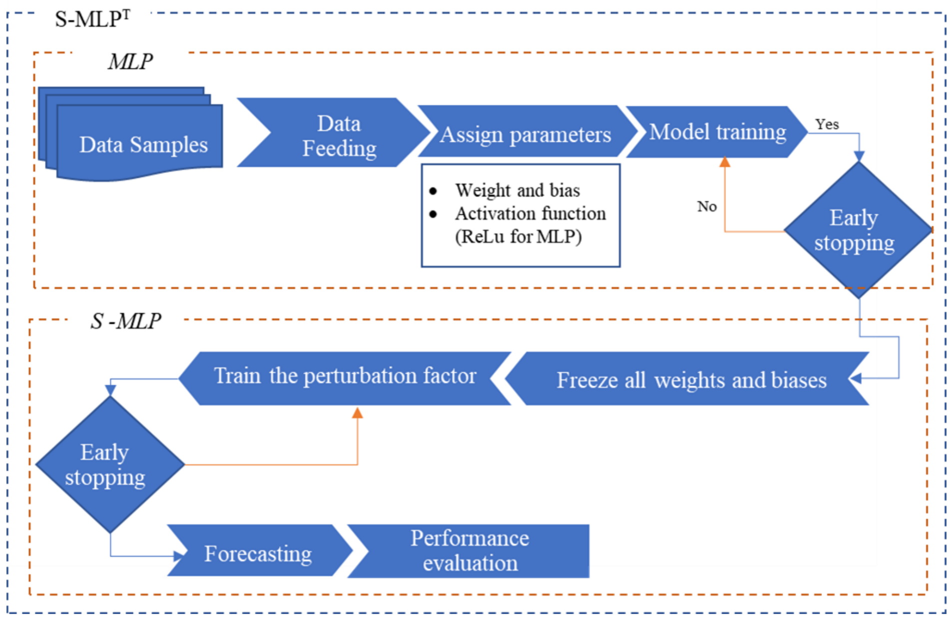

Figure 4. Structural overview of S-MLPT predictive model

Results and Discussion

The forecasting result MLP, S-MLP and S-MLP^T is tabulated in Table 1 which comprises of R^2 values of the predictive models for all the stock counters. From Table 1, it can be observed that MLP results in a positive value R^2 for 4065.KL which is 0.9749, whereas S-MLP and S-MLP^T R^2 values greater than 0.85 for all counters. Thus, this value indicates that the forecasting is quite accurate. However, R^2 value of MLP for 0166.KL and 4707.KL resulted a negative value, hence it is strongly suggested that the MLP has the worse fit [30].

Table 1. R^2 value of the MLP, S-MLP and S-MLP^T

Counter	R^2		
	MLP	S-MLP	S-MLP^T
0166.KL	**-3206**	0.9844	0.9858
2445.KL	**0**	0.8596	0.9924
4065.KL	0.9749	0.9756	0.9701
4707.KL	**-5×10^{11}**	0.9857	0.9905

In addition, Table 1 also shows that both S-MLP and S-MLPT outperformed MLP for all four counters. A comparative analysis was conducted by calculating the relative percentage improvement of S-MLP compared to MLP as the baseline model. The results indicated a 100% improvement for 0166.KL, 2445.KL, and 4707.KL counters. However, for the 4065.KL counter, S-MLP showed a slight improvement of 0.1%. Similarly, when calculating the relative percentage improvement of S-MLP^T compared to MLP, a 100% improvement can be observed for 0166.KL, 2445.KL, and 4707.KL counters. Based on these findings, it can be concluded that both S-MLP and S- MLP^T perform better than MLP.

Furthermore, an investigation was conducted to understand the reason s behind the superior performance of S-MLP compared to MLP in forecasting the 0166.KL, 2445.KL, and 4707.KL counters. Thus, the final cut off value of the training phase and the corresponding loss function values is identified and presented in Table 2. Notably, it is observed that the training of MLP was terminated earlier than that of S-MLP due to an early stopping mechanism. Besides, the loss function values of MLP were consistently higher than those of S-MLP across all counters. To further explore this phenomenon, the learning curves depicting the loss function during the training of both MLP and S-MLP are plotted in Figure 5.

Figure 5 reveals that both MLP and S-MLP reached a plateau after a certain number of epochs, leading to the activation of the early stopping mechanism. The introduction of stochastic processes into the sigmoid activation function enabled S-MLP to overcome these suboptimal points and converge towards improved local minima or even the global minimum, aligning with studies by Gulcehre *et al*. [18], Jay *et al*. [25], and Reddy & J.C. [40]. Consequently, it can be inferred from Table 2 and Figure 5 that S-MLP outperforms MLP due to its ability to overcome the drawback associated with MLP, which is being trapped in poor local minima or saddle points.

Table 2. Value of loss function and corresponding activation after final cut off training during early stopping mechanism

Counter	MLP		S-MLP	
	MSE-loss	No of epochs	MSE-loss	No of epochs
0166.KL	0.6120	20	0.0071	195
2445.KL	3.0703	19	0.1120	111
4065.KL	0.1019	39	0.0385	158
4707.KL	474.9989	28	9.1346	38

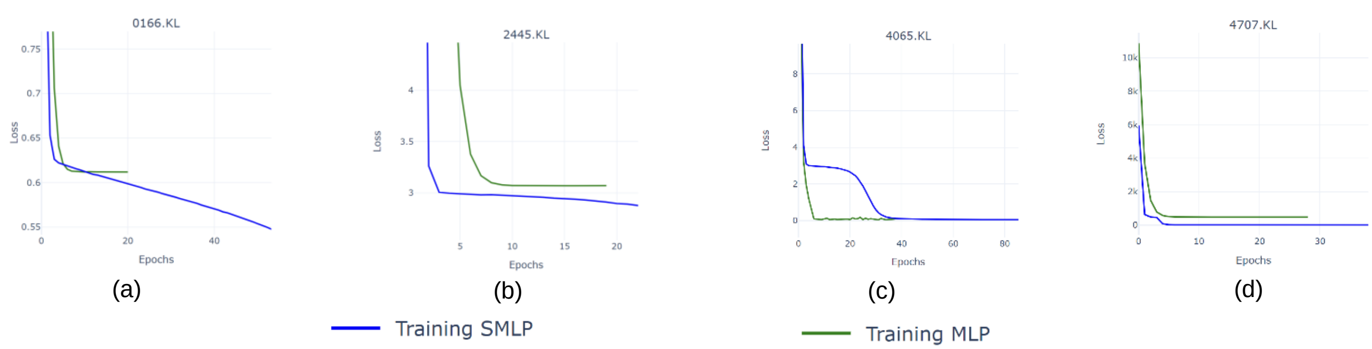

Figure 5. Loss function of training using MLP and S-MLP

Finally, to determine whether S-MLP^T had better accuracy than S-MLP, this study calculated the relative percentage improvement of S-MLP^T by using S-MLP as the reference model, and the result is recorded in Table 3. The table indicates that S-MLP^T showed an improvement in the accuracy of forecasting for all the counters except for 4065.KL. Hence, it can be concluded that, the accuracy of forecasting by S-MLP can be further improved by training the perturbation factor of the SAF after freezing the weight and bias parameters of trained MLP.

Table 3. Relative percentage improvement of S-MLP^T for all counters in comparison to S-MLP

Counter	0166.KL	2445.KL	4065.KL	4707.KL
Percentage improvement (%)	**0.14**	**15.45**	-0.57	**0.48**

Conclusions

In this manuscript, the research identified the reasoning on why S-MLP had better accuracy than MLP, which is because the incorporation of the gaussian process into the activation function allowed the loss function of the neural network to escape the poor local minima or saddle point. Moreover, this study attempted to improve the accuracy of forecasting of S-MLP by proposing S-MLP^T. Inspired by the work of Jay *et al.* [25], we first trained MLP and then froze the values of the weight and bias parameters. These frozen values were then loaded into S-MLP to train the perturbation factor of SAF using the backpropagation algorithm, optimized with SGD. The forecasting of one day ahead highest stock price of four counters listed in Bursa Malaysia, namely 0166.KL, 2445.KL, 4065.KL and 4707.KL was conducted using MLP, S-MLP and S-MLP^T. Our finding concludes that S-MLP^T outperforms both S-MLP and MLP interns in forecasting accuracy. This research achieved two objectives on investigating the learning curve of S-MLP and enhanced the accuracy of S-MLP by training the stochastic parameter via backpropagation. In future studies, it would be worth advancing S-MLP and S-MLP^T in portfolio optimization models which has been done by Ma *et al*. [41] and Solin *et al*. [42]. Apart from that, MLP is widely applied in climate fields such as forecasting rainfall, drought, hydrological processes (rainfall-runoff, groundwater evaluation). Hence, it is recommended to apply this model in various fields to validate the efficiency and performance of the model.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

The authors would like to acknowledge Universiti Malaysia Terengganu (UMT), Malaysia for the financial support of this research through the Talent and Publication Enhancement-Research Grant (UMT/TAPE-RG 2023/Vot 55463).

References

[1] Giles, C. L., Lawrence, S., & Tsoi, A. C. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning, 44*(1–2), 161. https://doi.org/10.1023/A:1010884214864

[2] Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence, 6*, 205–213. https://doi.org/10.1007/BF00126626

[3] Namdari, A., & Durrani, T. S. (2021). A multilayer feedforward perceptron model in neural networks for predicting stock market short-term trends. *Operations Research Forum, 2*(3), 1–30. https://doi.org/10.1007/s43069-021-00071-2

[4] Bhattacharjee, I., & Bhattacharja, P. (2019, December). Stock price prediction: A comparative study between traditional statistical approach and machine learning approach. In *2019 4th International Conference on Electrical Information and Communication Technology (EICT)* (pp. 1–6). IEEE. https://doi.org/10.1109/EICT48899.2019.9068850

[5] Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics, 2014*. https://doi.org/10.1155/2014/614342

[6] Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science, 132*, 1351–1362. https://doi.org/10.1016/J.PROCS.2018.05.050

[7] Kumar, D. A., & Murugan, S. (2013). Performance analysis of Indian stock market index using neural network time series model. *Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME 2013)*, 72–78. https://doi.org/10.1109/ICPRIME.2013.6496450

[8] Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications, 83*, 187–205. https://doi.org/10.1016/J.ESWA.2017.04.030

[9] Ling, H., Samarasinghe, S., & Kulasiri, D. (2016). Stochastic neural networks for modelling random processes from observed data. *Studies in Computational Intelligence, 628*, 83–107. https://doi.org/10.1007/978-3-319-28495-8_5

[10] Demirel, U., Cam, H., & Unlu, R. (2021). Predicting stock prices using machine learning methods and deep learning algorithms: The sample of the Istanbul Stock Exchange. *Gazi University Journal of Science, 34*(1), 63–82. https://doi.org/10.35378/GUJS.679103

[11] Anand, C. (2021). Comparison of stock price prediction models using pre-trained neural networks. *Journal of Ubiquitous Computing and Communication Technologies (UCCT), 3*(02), 122–134. https://doi.org/10.36548/jucct.2021.2.005

[12] Azizah, M., Irawan, M. I., & Putri, E. R. M. (2020, June). Comparison of stock price prediction using geometric Brownian motion and multilayer perceptron. In *AIP Conference Proceedings* (Vol. 2242, No. 1, p. 030016). AIP Publishing LLC. https://doi.org/10.1063/5.0008066/621859

[13] Chen, Q., Zhang, W., & Lou, Y. (2020). Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. *IEEE Access, 8*, 117365–117376. https://doi.org/10.1109/ACCESS.2020.3004284

[14] Hua, Y., Zhu, R., & Duan, Y. (2022, December). Construction of short-term stock price prediction algorithm based on MLP and CART Bagging ensemble learning. In *2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)* (pp. 371–376). IEEE. https://doi.org/10.1109/TOCS56154.2022.10015919

[15] Vincent, A. M. P., & Salleh, H. (2021). An investigation into the performance of the multilayer perceptron architecture of deep learning in forecasting stock prices. *Universiti Malaysia Terengganu Journal of Undergraduate Research, 3*(2), 61–68. https://doi.org/10.46754/umtjur.v3i2.205

[16] De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting, 22*(3), 443–473.

[17] Hendikawati, P. (2020, August). A survey of time series forecasting from stochastic method to soft computing. In *Journal of Physics: Conference Series* (Vol. 1613, No. 1, p. 012019). IOP Publishing. https://doi.org/10.1088/1742-6596/1613/1/012019

[18] Gulcehre, C., Moczulski, M., Denil, M., & Bengio, Y. (2016, June). Noisy activation functions. In *International Conference on Machine Learning* (pp. 3059–3068). PMLR. https://doi.org/10.48550/arxiv.1603.00391

[19] Liao, Z., & Wang, J. (2010). Forecasting model of global stock index by stochastic time effective neural network. *Expert Systems with Applications, 37*(1), 834–841. https://doi.org/10.1016/J.ESWA.2009.05.086

[20] Wang, J., Pan, H., & Liu, F. (2012). Forecasting crude oil price and stock price by jump stochastic time effective neural network model. *Journal of Applied Mathematics, 2012*. https://doi.org/10.1155/2012/646475

[21] Wang, J., Pan, H., Wang, Y., & Niu, H. (2015). Complex system analysis on voter stochastic system and jump time effective neural network of stock market. *International Journal of Computational Intelligence Systems, 8*(4), 787–795. https://doi.org/10.1080/18756891.2015.1061397

[22] Wang, J., & Wang, J. (2015). Forecasting stock market indexes using principal component analysis and stochastic time effective neural networks. *Neurocomputing, 156*, 68–78. https://doi.org/10.1016/J.NEUCOM.2014.12.084

[23] Mo, H., & Wang, J. (2018). Return scaling cross-correlation forecasting by stochastic time strength neural network in financial market dynamics. *Soft Computing, 22*, 3097–3109. https://doi.org/10.1007/s00500-017-2564-0

[24] Wang, J., & Wang, J. (2017). Forecasting stochastic neural network based on financial empirical mode decomposition. *Neural Networks, 90*, 8–20. https://doi.org/10.1016/J.NEUNET.2017.03.004

[25] Jay, P., Kalariya, V., Parmar, P., Tanwar, S., Kumar, N., & Alazab, M. (2020). Stochastic neural networks for cryptocurrency price prediction. *IEEE Access, 8*, 82804–82818. https://doi.org/10.1109/ACCESS.2020.2990659

[26] Vincent, A. M. P., & Salleh, H. (2023). Adaptation of stochasticity into activation function of deep learning for stock price forecasting. *Bulletin of Electrical Engineering and Informatics, 12*(6), 3780–3789.

[27] Wu, D., Wang, X., & Wu, S. (2022). Jointly modeling transfer learning of industrial chain information and deep learning for stock prediction. *Expert Systems with Applications, 191*, 116257. https://doi.org/10.1016/j.eswa.2021.116257

[28] Nikou, M., Mansourfar, G., & Bagherzadeh, J. (2019). Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms. *Intelligent Systems in Accounting, Finance and Management, 26*(4), 164–174. https://doi.org/10.1002/isaf.1459

[29] Zhang, J., Li, L., & Chen, W. (2020). Predicting stock price using two-stage machine learning techniques. *Computational Economics, 57*, 1237–1261. https://doi.org/10.1007/s10614-020-10013-5

[30] Alwosheel, A., van Cranenburgh, S., & Chorus, C. G. (2018). Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of Choice Modelling, 28*, 167–182. https://doi.org/10.1016/j.jocm.2018.07.002

[31] Kotsiantis, S. B., Kanellopoulos, D., & Pintelas, P. E. (2006). Data preprocessing for supervised learning. *International Journal of Computer Science, 1*(2), 111–117. https://doi.org/10.5281/zenodo.1082415

[32] Eelbode, T., Sinonquel, P., Maes, F., & Bisschops, R. (2021). Pitfalls in training and validation of deep learning systems. *Best Practice & Research Clinical Gastroenterology, 52*, 101712. https://doi.org/10.1016/j.bpg.2020.101712

[33] Sagir, A. M., & Sathasivan, S. (2017). The use of artificial neural network and multiple linear regressions for stock market forecasting. *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics*, 1–10.

[34] Yassin, I. M., Khalid, M. A., Herman, S. H., Pasya, I., Wahab, N. A., & Awang, Z. (2017). Multi-layer perceptron (MLP)-based nonlinear auto-regressive with exogenous inputs (NARX) stock forecasting model. *International Journal of Advanced Science, Engineering and Information Technology, 7*(3), 1098–1103.

[35] Masters, T. (1993). *Practical neural network recipes in C++*. Academic Press Professional, Inc.

[36] Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 249–256). JMLR Workshop and Conference Proceedings. https://proceedings.mlr.press/v9/glorot10a.html

[37] Vințe, C., & Ausloos, M. (2023). Portfolio volatility estimation relative to stock market cross-sectional intrinsic entropy. *Journal of Risk and Financial Management, 16*(2), 114. https://doi.org/10.3390/JRFM16020114

[38] Yang, D., & Zhang, Q. (2000). Drift-independent volatility estimation based on high, low, open, and close prices. *The Journal of Business, 73*(3), 477–492. https://doi.org/10.1086/209650

[39] Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science, 7*, e623. https://doi.org/10.7717/PEERJ-CS.623/SUPP-1

[40] Reddy, B. T. (2019). Prediction of stock market using stochastic neural networks. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)* ISSN, 2347–5552. https://doi.org/10.21276/IJIRCST.2019.7.5.1

[41] Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications, 165*, 113973.

[42] Solin, M. M., Alamsyah, A., Rikumahu, B., & Saputra, M. A. A. (2019, July). Forecasting portfolio optimization using artificial neural network and genetic algorithm. In *2019 7th International Conference on Information and Communication Technology (ICoICT)* (pp. 1–7). IEEE.