MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 1, pp. 176–186 (2025)

Neural network models with different input: An application on stock market forecasting

Ang A.[1], Nor M. E.[1], Rahman N. H. A.[2], Kamisan N. A. B.[3]

[1] *Department of Mathematics and Statistics, Faculty of Applied Science and Technology, Universiti Tun Hussein Onn Malaysia, Pagoh Edu Hub, 84600, Johor, Malaysia*

[2] *Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Lebuh Persiaran Tun Khalil Yaakob, 26300 Gambang, Pahang, Malaysia*

[3] *Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia*

(Received 15 October 2024; Revised 14 February 2025; Accepted 16 February 2025)

It is no doubt challenging to forecast the stock market accurately in reality due to the ever-changing market. Ever since Artificial Neural Networks (ANNs) have been recognized as universal approximators, they are extensively used in forecasting albeit not having a systematic approach in identifying optimal input. The appropriate number of significant lags of a time series corresponds to the optimal input in time series forecasting. Hence, this study aims to compare the effect of several approaches in determining the input lag for ANNs prior to stock market forecasting, based on the autocorrelation function, the partial autocorrelation function, the Box–Jenkins model and forward selection. The forecast performances of the ANNs were compared with benchmark models, namely the naïve and Box–Jenkins models, in terms of error magnitudes and trend change error. In this study, all ANNs were found to outperform the benchmark models such that the neural network model trained with lags selected from forward selection of lag 1 and lag 31 (ANN4) is the best model as it achieved the highest accuracy with the lowest mean absolute percentage error and mean absolute error. Contrary to expectations, all models performed poorly in forecasting the trend change of the stock series. The ANNs with different inputs are viable in quantitative stock market forecasting yet more research is required to better understand other trend change measurements and improve the performance of forecasting the trend change of stock market.

Keywords: *artificial neural networks; stock market forecasting; time series; input; forecast evaluation.*

2010 MSC: 92B20, 37M10 **DOI:** 10.23939/mmc2025.01.176

1. Introduction

Forecasting the stock market is a real-world financial problem spanning various fields, especially business, finance, and government. Stock trading is a way companies generate capital to fund and expand their businesses and pay off debts. Likewise, investors can gain profit from it in terms of dividends and take advantage of a growing economy. Stock market plays a part in stimulating commerce and the performance of the stock market can impact a nation's economy. Accurate prediction of stock values brings about better decision-making and financial planning, reduces risks and and ultimately maximizes profits [1]. However, stock market forecasting remains challenging due to a non-linear, highly dynamic property and any factors potentially affecting the stock market, such as politics, exchange rates, and catastrophes.

Nonetheless, it has received much attention from academia to develop statistical and technical modelling techniques as mentioned by [2] and [3]. For the past few decades, Artificial Neural Networks

This research was made possible by funding from FRGS Grant [FRGS/1/2019/STG06/UTHM/02/7] provided by the Ministry of Higher Education, Malaysia.

(ANNs) have been shown to perform effectively in forecasting the stock market due to their ability to learn and recognize new patterns [4]. Reference [5] applied the Multilayer Perceptron, Recurrent Neural Networks, Long Short-Term Memory, the Convolutional Neural Network, and the Autoregressive Integrated Moving Average (ARIMA) to predict the closing stock prices of the National Stock Exchange (NSE) of India and New York Stock Exchange (NYSE). All the NNs, each with an accuracy of more than 90%, outperformed ARIMA. Reference [6] suggested that ANN is superior in forecasting stock price changes on the Tehran Stock Exchange. Still, ANN cannot predict the direction change of stock value as well as Elman recurrent and linear regression models. Reference [7] compare the performance of three methods in stock forecasting in Indonesia. ARIMA and Autoregressive Fractionally Integrated Moving Average (ARFIMA) yield better predictions than ANN. ANN only gives better predictions in the training set and suggests better parameter tuning for an improved ANN model.

Hyperparameters in ANN must be tuned to achieve a better model. Some examples of hyperparameters are batch size, optimizer, learning rates, loss function, and activation functions [8]. The hyperparameters can be optimized through different methods, such as random or grid search [9]. Grid search explores all possible hyperparameter combinations to find the best model, while random search tests a random selection, making grid search more commonly used in many studies. Initially, a model is trained using a set of predefined hyperparameters, cross-validation is applied, and the optimal hyperparameter sets are identified afterward. Besides hyperparameter tuning, selecting potential inputs is one of the crucial steps that some researchers would tend to overlook in a modelling process. It is fundamental to the reliable performance of the model, especially in data-driven approaches such as ANNs. Relying merely on the network to determine the critical model inputs while feeding the network with a large number of inputs is considered suboptimal practice because irrelevant inputs will lead to model inaccuracy and computational inefficiency [10].

However, there are no standard procedures or systematic ways accepted in the literature to identify an appropriate set of inputs in ANN modeling, as mentioned by [11]. Hence, significant input selection methodology for ANNs is still being actively investigated. In time series forecasting, the network inputs are the number of lagged observations of the series, and oftentimes, trial-and-error is employed to determine the inputs, as raised by [12] and [13]. The drawback of such practice is that it raises questions about the optimality of ANN models if they are, by chance, achieved, as those studies cannot be replicated. Reference [14] discovered that the ANN model achieved a higher accuracy when the time lags from Box–Jenkins were adopted as ANN inputs rather than running exhaustive numerical simulations in predicting water consumption. Reference [15] explored ten input selection methods for linear and nonlinear models to select temporal lags in predicting river flow series from hydroelectric plants and found that the neural networks performed better with forward selection, but it is computationally intensive for large sample sizes.

No consistent results can be observed regarding the issue of determining an appropriate number of inputs in ANN modelling, which is in contrast to statistical modelling such as Box–Jenkins, as there are usually steps and tests to be taken to construct a model. To the best of our knowledge, no article focused on comparing the effectiveness of input selection methods on univariate stock price forecasting. Hence, the study's first objective was to determine the inputs of ANNs based on the autocorrelation function (ACF), partial autocorrelation function (PACF), Box–Jenkins model, and forward selection. This study also aimed to forecast stock market changes using ANNs and benchmark models, namely, Box–Jenkins and naïve models, and to compare their forecast performances in terms of error magnitudes and trend change error.

2. Materials and methods

The data used in this study are 2558 historical daily stock closing prices for one of the gas infrastructure and centralized utility companies which was retrieved from yahoo finance [16]. The historical data were partitioned into training and testing sets before modeling. The training data were used to train the model so that the characteristics of the data could be captured while estimating the parameters and minimizing the error. The testing data were treated as unseen data to evaluate how well the model performs or generalizes with fresh data. In this study, the training set comprised data from 3rd January 2011 to 16th April 2021. This training data was used to forecast four-week stock prices for 18 days, from 19th April 2021 to 17th May 2021, which was also the validation period to evaluate the forecast accuracy of each model. The analysis was carried out using Excel, R, and Python software. Preliminary data analysis was conducted before data partition to check the stationarity of the time series by plotting the time series plot and autocorrelation plot. A non-stationary time series exhibits certain trends, seasonality, or cyclic fluctuations with a slow-decaying ACF.

2.1. Artificial neural networks model

ANNs are nonlinear models. ANNs are renowned for their self-adaptive and generalization features, in which prior assumptions are not required during modelling as they can learn from experience, the relationships among data, when data are fed into the networks as mentioned by [12]. Since ANN emulates the biological neural system, an ANN is composed of a bunch of interconnected nodes or neurons arranged in layers. Generally, the network is composed of input, hidden and output layers, where the output of a layer serves as the input of the next layer respectively. This study considered the classical, popular Multilayer Perceptron (MLP) with a single hidden layer. MLP comprises one input layer, one hidden layer, and one output layer, as depicted in Figure 1.

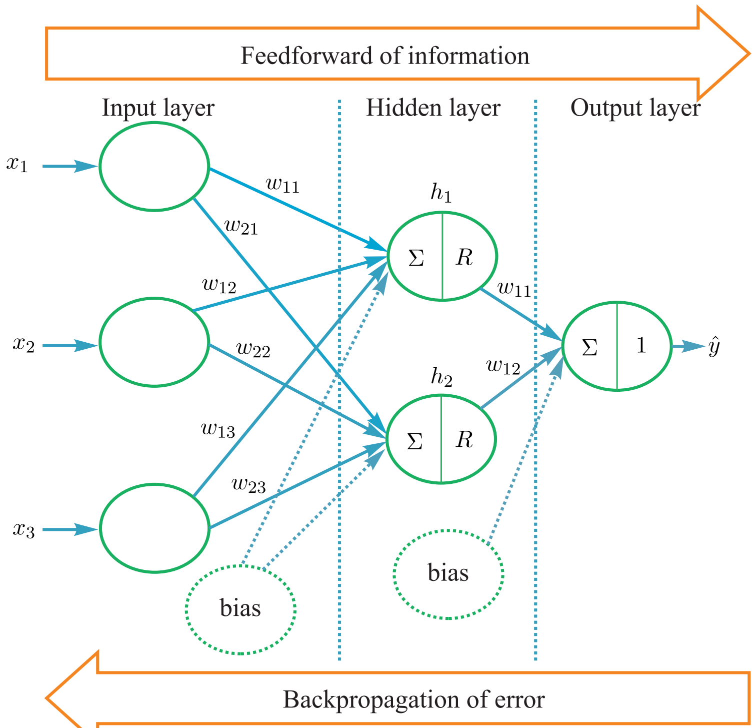

Fig. 1. Multilayer Perceptron with a single hidden layer.

Since MLP is a feedforward network, the data flows forward from the input layer to the output layer. Every neuron or node in a layer is fully connected to every neuron in the next layer. The input neurons in the input layer receive input from the data, which, in this case, is the lagged values of the stock prices. The values obtained are multiplied with weights and a bias is added to the summation of inputs and weights, which are then served as the input of the next layer. The weights control the connection strength between two neurons. The larger the weight, the higher the influence of the input has on the output. Bias is a constant that is not influenced by the previous layer but ensures a neuron's activation even if all the inputs are zero. The signal of each neuron obtained from the linear function of each node in the hidden layer requires an activation function to be transmitted to the output layer. In this study, the rectified linear (ReLU) unit was chosen as the activation function since ReLU is a solution to the vanishing gradient problem that Sigmoid experiences, as mentioned by [9]. Reference [17] mentioned linear activation function is common for the output layer. The output generated by the activation function is in the range of $[0,1]$ or $[-1,1]$. The outputs of the hidden layer are then again multiplied with weights along with an additional bias. Then, the output neuron is activated by an activation function to finally produce an output which in this case, is the predicted stock prices. The output $\hat{y}$ can be written as in Eq. (1), where denotes the activation function, ϕ_0 denotes the bias, w_{oi} denotes the weight from i^{th} neuron in the hidden layer to o^{th} neuron in the output layer and h_i denotes outputs of the hidden layer, which serve as inputs of the output layer neuron,

$$\hat{y} = \phi\left(b_0 + \sum w_{oi} h_i\right). \tag{1}$$

Since neural network training is an unconstrained nonlinear minimization problem, the network weights are optimized to minimize the overall mean or total squared error between the actual and expected values for all output neurons over all training examples [12]. The backpropagation algorithm is commonly used to train the network, and it works by propagating the error back through the network while optimizing or readjusting the weights in a manner that minimizes the error e which can be written as in Eq. (2), where y denotes the actual value, denotes the expected value and n denotes the number of training examples. One "epoch" consists of a cycle of feedforward and backpropagation. Iterations are conducted until a reasonable accuracy is obtained. This study applied the grid search method to automatically tune the hyperparameters, consisting of the number of hidden neurons, optimizer, batch size, and epochs,

$$e = \sum_n (y - \hat{y})^2. \tag{2}$$

2.1.1. Data normalization

According to [18] the purpose of normalizing the data before feeding the neural network with input is to improve error estimations and speed up the learning process of the network. This study used min-max normalization in which values were normalized to a range of zero to one [19]. The normalized value y_t' can be computed as in Eq. (3), where $\min(y_t)$ and $\max(y_t)$ is the minimum and maximum of y_t over the range of data,

$$y_t' = \frac{y_t - \min(y_t)}{\max(y_t) - \min(y_t)}. \tag{3}$$

2.1.2. Input selection methods of artificial neural networks model

Reference [12] defined in time series forecasting that the inputs are the past observations of the time series of its own variable, otherwise known as lag, while the outputs are the future values. It is difficult to determine how many lagged values to include as additional lagged values will have no significant effect on the output time series as mentioned by [10]. This study selected the number of lags from the autocorrelation function (ACF), partial autocorrelation function (PACF), Box–Jenkins model, and forward selection during the first phase of ANN modelling.

Autocorrelation and partial autocorrelation function. ACF can be defined as the correlation between two observations of the same series at a certain lag. The ACF for an observation and its lagged value includes both the direct and indirect correlations, in which the indirect correlations are a linear function of the correlation. On the other hand, PACF at lag k is the resulting conditional correlation after omitting the indirect correlations between the observation and its lagged value as defined by [20]. The significant lags that exceed the threshold, as observed from the plots of ACF and PACF, were assigned as the input of ANNs.

Lag in Box–Jenkins model. In this study, Box–Jenkins analysis was also used to identify relevant lags. This was done by expanding the selected Box–Jenkins model to obtain the respective lags from each term [14]. The lags identified by the Box–Jenkins model that yielded the smallest Akaike Information Criterion (AIC), corrected AIC (AICc), and Bayesian Information Criterion (BIC) were adapted as ANNs input.

Forward selection. According to [21] forward selection is one of the wrapper method where individual candidate variables are selected one at a time. In this study, each variable represents each lag of the time series. The maximum lag length $k_{\max}$, the included was based on the default maximum lag length of ACF and PACF functions in R software, which can be computed as in Eq. (4), where n denotes the number of training examples, and M denotes the number of time series,

$$k_{\max} = 10 \log \tfrac{n}{M}. \quad (4)$$

Upon computation, the maximum lag length used in this study was 34. The forward selection algorithm started by training the ANN model with v single input variable, from lag 1 to lag 34 subsequently, while finding the best single input or lag that minimized the errors. Then, the training of $v - 1$ bivariate ANN model was continued while adding the candidate lag to the input variable chosen previously. Finally, the iteration was stopped when the combination of lags with the addition of another lag did not further minimize the errors or improve the performance of the ANN model.

2.2. Box–Jenkins model

Box–Jenkins, also referred to as ARIMA, is a combination of autoregressive (AR) and moving average (MA) models integrating the process of differencing to make the time series stationary as defined by [22]. The general form of an ARIMA model is ARIMA(p, d, q), where p denotes the order of autoregressive, q denotes the order of the moving average, and d denotes the order of differencing. The general non-seasonal Box–Jenkins model can be written as in Eq. (5), where B is the backshift operator, ϕ denotes non-seasonal autoregressive parameter of order p, θ denotes non-seasonal moving average parameter of order q, a_t denotes independent normal errors with zero mean and constant variance,

$$\phi_p(B)(1 - B)^d y_t = \theta_q(B)a_t. \quad (5)$$

There are three main phases in modeling Box–Jenkins: model identification, parameter estimation and residual diagnostics. If the data exhibit inconsistent variation, Box–Cox transformation can be applied to stabilize the variance. The process of differencing can be performed to remove the trend or stabilize the mean of the series. Once the series is confirmed to be stationary with stable variance and mean, ACF and PACF can be plotted to determine tentative ARIMA models. If the estimated parameters of the tentative model are statistically significant with p-value less than the significance level of 0.05, the tentative model can then be fitted to the data. Lastly, residual diagnostics should be conducted. The model is adequate if the model yields uncorrelated residuals with zero mean [22]. The residuals are independent if the p-values from the Ljung–Box test exceed the 0.05 significance level. An unbiased forecast should yield a residual series that is centered and close to zero with no trend, indicating that the residuals have a zero mean.

Model selection criteria. Besides serving as one of the benchmark models, the respective lags from the selected Box–Jenkins model with the smallest AIC, AICc, and BIC values were assigned as the ANN model's input. Criteria such as AIC, AICc and BIC are useful in Box–Jenkins model identification [22]. Smaller value of the criteria indicates a better model. AIC, AICc and BIC can be expressed as in Eqs. (6)–(8) respectively, where L denotes the likelihood of the data, k denotes the lag of a time series and n denotes the number of observations,

$$\text{AIC} = -2\log(L) + 2(p + q + k + 1), \quad (6)$$

$$\text{AICc} = \text{AIC} + \tfrac{2(p+q+k+1)(p+q+k+2)}{n-p-q-k-2}, \quad (7)$$

$$\text{BIC} = \text{AIC} + [\log(n) - 2](p + q + k + 1). \quad (8)$$

2.3. Naïve model

The naïve model usually serves as a basis to compare with other complex models since it is the simplest form of the forecasting model. It has the potential to perform well for economic and financial time series especially when the data follow a random walk behaviour, so, the model can also be referred as a random walk model [22]. The forecast for the next period is the current actual value y_t, which can be written as in Eq. (9),

$$\hat{y}_{t+1} = y_t. \quad (9)$$

2.4. Measures of accuracy

2.4.1. Error Magnitudes

Mean forecast error (MFE) in Eq. (10), mean absolute percentage error (MAPE) in Eq. (11), and mean absolute error (MAE) in Eq. (12), were used in this study to evaluate the quantitative forecast performance of the models. Based on the equations, denotes the actual value, denotes the forecasted value and N denotes the total forecast periods,

$$\text{MFE} = \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t), \tag{10}$$

$$\text{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100\%, \tag{11}$$

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|y_t - \hat{y}_t|. \tag{12}$$

2.4.2. Trend change error

Trend change measurement is equally crucial in stock market forecasting because it indicates when the trend will change from a positive growth rate to a negative growth rate or the other way round, which would come in handy for market players. Trend change error refers to the error made when the forecasting model attempts to predict a change in the trend of a data series [23]. This study adopted the definitions of trend change error as shown in Eq. (13) and Eq. (14), where Z represents the first future value of the series,

$$y_{t-2} < y_{t-1} < y_t \quad \text{and} \quad \begin{cases} Z < y_t \equiv \text{Downturn}, \\ Z \geqslant y_t \equiv \text{No Downturn}; \end{cases} \tag{13}$$

$$y_{t-2} > y_{t-1} > y_t \quad \text{and} \quad \begin{cases} Z > y_t \equiv \text{Upturn}, \\ Z \leqslant y_t \equiv \text{No Upturn}. \end{cases} \tag{14}$$

3. Results and discussion

3.1. Stationarity detection

Figure 2 shows the time series plot of the daily stock prices from 3rd January 2011 to 17th May 2021 and the vertical line marks the division of 2540 training data and 18 testing data. Based on Figure 2, the time series plot exhibits certain trend that overall increases from 2011 and decreases gradually around mid-2014 with potential cyclic patterns since the rises and falls of the series are not fixed at regular intervals. This also indicates that the variation of the series might not be constant too. The slow decaying of the ACF as the lags increase indicates the presence of the trend (refer Figure 3). Thus, the stock price series is not stationary. Data was transformed using Box–Cox transformation, in which the value of lambda was determined from BoxCox.lambda() function in R, which minimizes the coefficient of variation of the series is 1.0056, followed by the first order of regular differencing prior to Box–Jenkins modeling, so that the series is stationary. Likewise, input data was normalized before training ANNs to ensure even data distribution and effective modeling process.

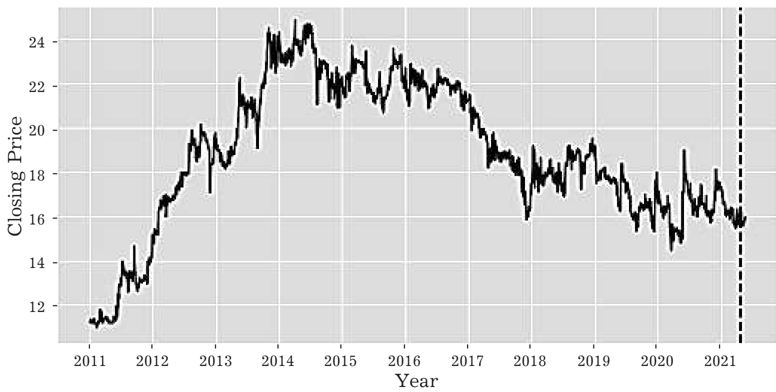

Fig. 2. Time series plot of daily stock prices.

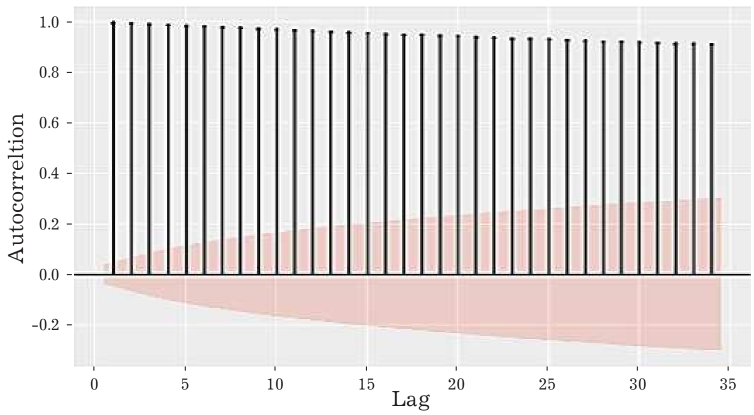

Fig. 3. ACF of daily stock prices.

3.2. Lag selection

Referring to the ACF and PACF plots in Figure 4 and Figure 5 respectively, the significant lags exceeding the threshold are lag 1 and lag 3 in both plots. While modeling Box–Jenkins, ARIMA(1,1,2) was found to have the lowest AIC and AICc whereas ARIMA(0,1,1) produced the lowest BIC among other tentative models after making sure that both models have significant parameters with uncorrelated, zero mean residuals. This means that both models are adequate to generate forecast. Thus, the lags resulted from both AR and MA term of ARIMA(1,1,2) was obtained through the expansion of model as shown in Eq. (15), where the estimated parameters are $\phi_1 = 0.9059$, $\theta_1 = -1.0249$ and $\theta_2 = 0.0823$,

$$(1 - \phi_1 B)(1 - B)y_t = (1 - \theta_1 B - \theta_2 B^2)a_t,$$

$$y_t = y_{t-1} + 0.9059y_{t-1} - 0.9059y_{t-2} + 1.0249a_{t-1} - 0.0823a_{t-2} + a_t. \tag{15}$$

Similarly, the lags resulted from both AR and MA term of ARIMA(0,1,1) was obtained through the expansion of model as shown in Eq. (16), where the estimated parameter is $\theta_1 = 0.1185$,

$$(1 - B)y_t = (1 - \theta_1 B)a_t,$$

$$y_t = y_{t-1} - 0.1185a_{t-1} + a_t. \tag{16}$$

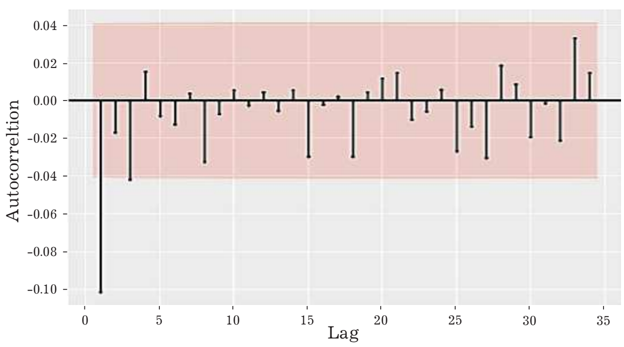

Fig. 4. ACF of differenced series.

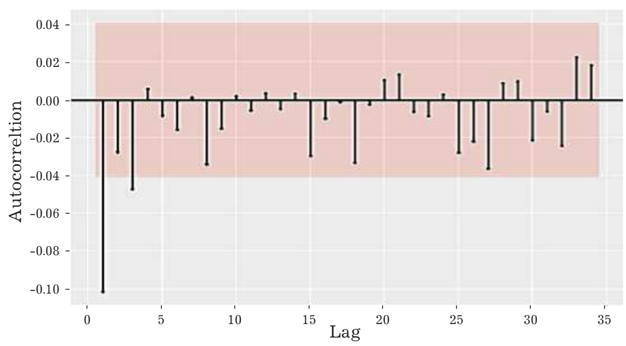

Fig. 5. PACF of differenced series.

For forward selection, the subset of lags included to train the base model of ANN which returned the lowest error was chosen. There are no strict rules for selecting hyperparameters for neural networks. Thus, the hyperparameters set for the base model are: 15 hidden neurons, batch size of 32, adaptive moment estimation (Adam) as optimizer, rectified linear unit activation function (ReLU) for the hidden layer and a linear activation function for the output layer. Adam was chosen as the optimizer for the base model because Adam is an optimization algorithm that has the properties of adaptive gradient and root mean square propagation optimization algorithms, which make it computationally efficient and suitable for noisy problems [24].

The first round of iteration of forward selection consisted of training the base model with one lag each time from lag 1 to lag 34 in ascending order and the base model trained with lag 1 was found to return the lowest error among other single lags. So, lag 1 was included during the second round of iterations. The second round of iteration was completed by training the base model with two lags, starting from lag 2 to lag 34, each time including lag 1. Since the base model with lag 1 and lag 31 returned a lower error than that with lag 1, a third round of iteration was run, three lags each time. Lag 1 and lag 31 were included in each training process followed by an additional lag from the group of remaining candidate lags. This time, the inclusion of a third lag did not further minimize the error, thus, the iteration of forward selection was stopped. Therefore, the lags obtained from forward selection are lag 1 and lag 31.

Table 1. Lag selection.

Method	Selected lag	ANN model
ARIMA(0,1,1)	1	ANN1
ARIMA(1,1,2)	1,2	ANN2
ACF and PACF	1,3	ANN3
Forward Selection	1,31	ANN4

Table 1 lists the selected lags from each method. Since certain methods yielded the same lags, four ANN models were formed.

3.3. Model comparison

Based on Table 2, in terms of error magnitudes, all models have very good forecast performance with very low MFE, MAPE and MAE that approach zero. All values are close with no significant difference. One plausible reason is that the stock data volatility is not that high, with no dramatic highs and lows every other day. However, the stock price series is clearly nonstationary due to trends. Since positive MFE means that a model has under-forecasted on average, it can be said that each ANN has slightly under-forecasted the four-week stock prices on average. Although the MFE of ANNs for testing set each is higher than the benchmark models, still, ANNs in general have better forecast performance with lower test error magnitudes of MAPE and MAE than the benchmark models used in this study. The finding agrees with [10] that ANNs have outperformed linear models in forecasting stock prices and have better generalization ability. Moreover, the forecasted pattern of Box–Jenkins is indeed more directional, accounting for a linear pattern instead, whereas ANNs have displayed curves that deviate from exact straight lines (refer to Figure 6). The finding that ANN is capable of capturing the nonlinear mappings of inputs and output well is supported by previous studies [5,6]. All ANN models have shown an overall curve of decreasing trend, thus, followed the actual series more closely, especially within the period of 27$^{\text{th}}$ April 2021 to 11$^{\text{th}}$ May 2021. However, the ANNs failed to forecast the values of the peaks on 23$^{\text{rd}}$ April and 17$^{\text{th}}$ May as well as the trough on 5$^{\text{th}}$ May 2021 accurately (refer to Figure 6).

Table 2. Forecast performance of each model.

Model	MFE	MAPE	MAE	Trend change error
Naïve	0.0111 (2)	0.6997 (5)	0.1111 (5)	100 (2)
ARIMA(1,1,2)	−0.0201 (3)	0.7958 (7)	0.1261 (7)	80 (1)
ARIMA(0,1,1)	0.0037 (1)	0.7155 (6)	0.1136 (6)	100 (2)
ANN1	0.0542 (6)	0.6513 (3)	0.1038 (3)	80 (1)
ANN2	0.0614 (7)	0.6496 (2)	0.1035 (2)	80 (1)
ANN3	0.0239 (4)	0.6685 (4)	0.1063 (4)	80 (1)
ANN4	0.0515 (5)	0.6426 (1)	0.1023 (1)	80 (1)

(): accuracy ranking

In terms of forecasting trend change, all models did not achieve satisfactory results because each model had a trend change error exceeding 50%. This result indicates that it is indeed difficult to properly predict turning points and that the definition adopted might not be suitable for the study. No downturn is among the types of trend changes forecast correctly by ARIMA(1,1,2). This is because the model has displayed a gradually increasing trend across the forecast period. On the other hand, no upturn is forecasted correctly by all ANNs because all models have displayed a gradual decreasing trend across the forecast period. Naïve and ARIMA(0,1,1) have failed to capture any trend change as both models show a horizontal straight line with no trend (refer to Figure 6).

Among the four ANN models, the fourth ANN model, trained using lag 1 and lag 31, outperformed the other three ANNs with the lowest test error magnitudes. The ANN2, which was trained using lag 1 and lag 2, is ranked second in terms of MAPE and MAE but last in terms of MFE. ANN4 which adopted the lags obtained from forward selection, which is considered a wrapper method that has slightly better performance than other ANNs, and this finding matches those observed in [15], where they concluded that the wrapper method is the ideal choice for neural networks in selecting lags to forecast river flow. The flip side of forward selection is that it is more time-consuming than other methods since multiple iterations had to be run to find the best combination of lags. However, the selection is optimized and tailored to the specific model for the neural network's learning process. Besides, it was also found that ANNs with lags adopted from Box–Jenkins analysis, which are ANN1 and ANN2, performed better than the Box–Jenkins models. This finding is also consistent with [14]. Each ANN model was trained using only certain lags, yet they were able to perform well. Since the ANNs achieved varying levels of accuracy with different lags, selecting proper lags as ANN inputs can improve ANN's accuracy. These results have further supported the idea that stock prices can be forecast and historical stock prices influence the stock prices within the forecast period.

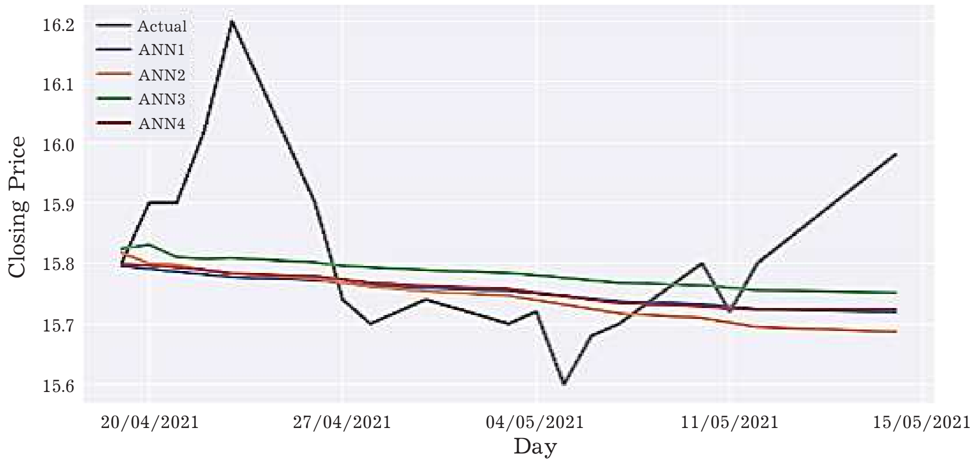

Fig. 6. Actual and forecasted stock price values using ANNs, Box–Jenkins and naïve approaches.

4. Conclusion

This study has confirmed the existing theory of previous research that ANNs have the inherent capability for accurately capturing nonlinear mappings of inputs to outputs. Overall, ANNs with different inputs outperformed other linear benchmark models in forecasting stock prices. Besides, the findings of this study have demonstrated that the accuracy of ANNs improved with proper input selection methods as measured by quantitative error metrics. ANN4 was found to be the best model in this study; that is, the ANN model, which adopted lags from forward selection achieved the best forecast performance among other ANNs in terms of MAPE and MAE. Despite that, it is unexpected that each benchmark model had a lower MFE than the ANNs, with ARIMA(0,1,1) recording the lowest MFE. Also, not all models may not be practical for predicting turning points, as their performance in forecasting trend changes in the stock price series was not encouraging. However, this result is not surprising as predicting trend change for volatile data such as the stock market is remarkably challenging compared to predicting the value. This is one of the reasons trend change error is less popular among researchers. However, it remains valuable to have information on forecast performance regarding trend change in addition to error magnitude. Besides, different inputs yielded close but different results in this study; it would be interesting to further assess the effects of other advanced input selection methods, such as mutual or partial mutual information.

[1] Phuoc T., Anh P. T. K., Tam P. H., Nguyen C. V. Applying machine learning algorithms to predict the stock price trend in the stock market – The case of Vietnam. Humanities and Social Sciences Communications. **11**, 393 (2024).

[2] Bustos O., Pomares-Quimbaya A. Stock market movement forecast: A Systematic review. Expert Systems with Applications. **156**, 113464 (2020).

[3] Dokuchaev N. On statistical indistinguishability of complete and incomplete discrete time market models. Decisions in Economics and Finance. **46**, 461–475 (2023).

[4] Rodriguez F. S., Norouzzadeh P., Anwar Z., Snir E., Rahmani B. A machine learning approach to predict the S&P 500 absolute percent change. Discover Artificial Intelligence. **4**, 8 (2024).

[5] Hiransha M., Gopalakrishnan E. A., Menon V. K., Soman K. P. NSE Stock Market Prediction Using Deep-Learning Models. Procedia Computer Science. **132**, 1351–1362 (2018).

[6] Naeini M. P., Taremian H., Hashemi H. B. Stock market value prediction using neural networks. 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM). 132–136 (2010).

[7] Tanuwijaya R., Saputra K. V. I., Ferdinand F. V. Comparison of ARIMA, ARFIMA and Artificial Neural Network Models for Indonesian Stocks Market. ICIC Express Letters, Part B: Applications. **14** (10), 1095–1106 (2023).

[8] Kalliola J., Kapočiūtė-Dzikienė J., Damaševičius R. Neural Network Hyperparameter Optimization for Prediction of Real Estate Prices in Helsinki. PeerJ Computer Science. **7**, e444 (2021).

[9] Rahman N. H. A., Yin C. H., Zulkafli H. S. Activation functions performance in multilayer perceptron for time series forecasting. AIP Conference Proceedings. **3123** (1), 070001 (2024).

[10] Bowden G. J., Dandy G. C., Maier H. R. Input determination for neural network models in water resources applications. Part 1 – Background and methodology. Journal of Hydrology. **301** (1–4), 75–92 (2005).

[11] Sánchez-Sánchez P. A., García-González J. R., Coronell L. H. P. Encountered Problems of Time Series with Neural Networks: Models and Architectures. Recent Trends in Artificial Neural Networks – from Training to Prediction. 1–17 (2019).

[12] Zhang G., Patuwo B. E., Hu M. Y. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting. **14** (1), 35–62 (1998).

[13] Pandey V. S., Bajpai A. Predictive Efficiency of ARIMA and ANN Models: A Case Analysis of Nifty Fifty in Indian Stock Market. International Journal of Applied Engineering Research. **14** (2), 232–244 (2019).

[14] BuHamra S., Smaoui N., Gabr M. The Box–Jenkins analysis and neural networks: prediction and time series modelling. Applied Mathematical Modelling. **27** (10), 805-815 (2003).

[15] Siqueira H., Macedo M., Tadano Y. d. S., Alves T. A., Stevan S. L. (Jr.), Oliveira D. S. (Jr.), et al. Selection of Temporal Lags for Predicting Riverflow Series from Hydroelectric Plants Using Variable Selection Methods. Energies. **13** (16), 4236 (2020).

[16] `https://finance.yahoo.com`

[17] Sarle W. S. Neural Networks and Statistical Models. Proceedings of the Nineteenth Annual SAS Users Group International Conference, Cary, NC: SAS Institute, USA. 1538–1550 (1994).

[18] Demir S., Sahin E. K. The effectiveness of data pre-processing methods on the performance of machine learning techniques using RF, SVR, Cubist and SGB: a study on undrained shear strength prediction. Stochastic Environmental Research and Risk Assessment. **38**, 3273–3290 (2024).

[19] Witten I. H., Frank E., Hall M. A., Pal C. J. Data Mining Practical Machine Learning Tools and Techniques. Elsevier Science; 4th edition (2016).

[20] Cowpertwait P. S. P., Metcalfe A. V. Introductory Time Series with R. Biometrics. **67** (1), 328–329 (2011).

[21] May R., Dandy G., Maier H. Review of Input Variable Selection Methods for Artificial Neural Networks. Artificial Neural Networks - Methodological Advances and Biomedical Applications. Vol. 2 (2011).

[22] Hyndman R., Athanasopoulos G. Forecasting: Principles and Practice. Otexts; 2nd edition (2018).

[23] Haw L. K., Nor M. E., Shab N. F. M. The Impact of Government Policy Changes on Stock Market Forecasting. Proceedings of the 8th International Conference on the Applications of Science and Mathematics. 301–315 (2023).

[24] Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. Conference Paper at the 3rd International Conference for Learning Representations, San Diego (2015).

Моделі нейронної мережі з різними вхідними даними: застосування до прогнозування фондового ринку

Анг А.[1], Нор М. Е.[1], Рахман Н. Х. А.[2], Камісан Н. А. Б.[3]

[1]*Кафедра математики та статистики, факультет прикладних наук і технологій, Університет Тун Хуссейн Онн Малайзія, Пагох Еду Хаб, 84600, Джохор, Малайзія*
[2]*Центр математичних наук Університету Малайзії Паханг Аль-Султан Абдулла, Лебух Персіаран Тун Халіл Якоб, 26300 Гамбанг, Паханг, Малайзія*
[3]*Кафедра математичних наук, факультет природничих наук, Технологічний університет Малайзії, 81310 Джохор-Бару, Джохор, Малайзія*

Безсумнівно, складно точно прогнозувати фондовий ринок у реальності через ринок, який постійно змінюється. З тих пір, як штучні нейронні мережі (ANN) визнано універсальними апроксиматорами, вони широко використовуються в прогнозуванні, хоча й не мають системного підходу до визначення оптимальних вхідних даних. Відповідна кількість значущих затримок часового ряду відповідає оптимальним вхідним даним в прогнозування часового ряду. Таким чином, це дослідження спрямоване на порівняння ефекту кількох підходів у визначенні вхідної затримки для ANN перед прогнозуванням фондового ринку на основі функції автокореляції, часткової функції автокореляції, моделі Бокса–Дженкінса та прямого вибору. Прогнозовані показники ANN порівнювалися з еталонними моделями, а саме наївними моделями та моделями Бокса–Дженкінса, з точки зору величини похибки та похибки зміни тренда. У цьому дослідженні було виявлено, що всі ANN перевершують еталонні моделі, тому модель нейронної мережі, яка навчена із затримками, що вибрані з прямого вибору затримки 1 і затримки 31 (ANN4), є найкращою моделлю, оскільки вона досягла найвищої точності з найнижчим середнім абсолютним відсотком похибки і середньою абсолютною похибкою. Всупереч очікуванням, усі моделі показали погані результати в прогнозуванні зміни тренду ряду акцій. ANN з різними вхідними даними є життєздатними в кількісному прогнозуванні фондового ринку, але необхідні додаткові дослідження, щоб краще зрозуміти інші вимірювання зміни тренда та покращити ефективність прогнозування зміни тренда фондового ринку.

Ключові слова: *штучні нейронні мережі; прогнозування фондового ринку; часові ряди; введення; оцінка прогнозу.*