



# A Multilayer Feedforward Perceptron Model in Neural Networks for Predicting Stock Market Short-term Trends

Alireza Namdari<sup>1</sup> · Tariq S. Durrani<sup>2</sup>

Received: 4 July 2020 / Accepted: 26 May 2021  
© The Author(s) 2021

## Abstract

Stock market prediction is important for investors seeking a return on the capital invested, though this prediction is a challenging task, due to the complexity of stock price time-series. This task can be performed by conducting two primary analyses: fundamental and technical. In this paper, we examine the predictability of these two analyses using a multilayer feedforward perceptron neural network (MLP) and determine whether MLP is capable of accurately predicting stock market short-term trends. We utilize stock prices (2013 Mar – 2018 Jun) and twelve financial ratios of technology companies selected through a feature selection preprocess. Our model uses self-organizing maps (SOMs) for clustering the historical prices and produces a low-dimensional discretized representation of the input space. The best results are obtained through hyper-parameter optimizations using a three-hidden layer MLP. The models are integrated using a nonlinear autoregressive structure with exogenous input (NARX). We find that the hybrid model successfully predicts the short-term stock trends. The hybrid model yields the greatest directional accuracy (70.36%) as compared to fundamental and technical analyses (64.38% and 62.85%) and state-of-the-art models. The results indicate that the market is not fully efficient. Our model will be useful to practitioners seeking investing and trading opportunities and others interested in the study of financial markets.

**Keywords** Multilayer feedforward perceptron · Hyper-parameter Optimization · Self-organizing maps · Feature selection · Data discretization · Stock market

---

✉ Tariq S. Durrani  
durrani@strath.ac.uk

Alireza Namdari  
an354920@wne.edu

<sup>1</sup> Department of Industrial Engineering and Engineering Management, Western New England University, Springfield, MA 01119, USA

<sup>2</sup> Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow Scotland, UK

## 1 Introduction

The prediction of the stock market performance is an important facet of financial markets due to the large amounts of money involved. This predictability has been extensively studied, and various hypotheses have been applied, such as efficient market hypothesis (EMH), which states that the stock market is not predictable. Instead, EMH suggests that the stock market is efficient, and therefore, it is not possible to outperform the market [1]. Fama, an American economist, studied EMH and conducted empirical works in the 1970s [2]. After Fama presented his study called “Efficient capital markets: A review of the theory and empirical work,” many responded with research papers either in support or against EMH [1–6, ]. For the people that criticize the EMH, they believe the stock market prices are unreliable and mispriced as a result of noises caused by the momentum traders, speculators, and insiders. Fundamental and technical are two primary analyses for making investment decisions in stock markets.

### 1.1 Fundamental Analysis

Fundamental analysis assumes the stock market is not perfectly efficient; therefore, it aims to determine the intrinsic value of the security and analyze it against the actual present value. This ensures that the mispriced securities are detected [7]. Fundamental analysis supports the idea that over time, stock prices will correct themselves. As a result, many investors *go long* on the undervalued securities while *going short* on the overvalued securities to balance out the prices. The fundamentalists use financial statements to conduct their analysis using the debt-to-equity ratio (DER), market-to-book ratio (MBR), gross profit margin (GPM), and quick ratio (QR) [8]. Furthermore, the financial ratios of companies within the same sector are weighed against each other to conclude an adequate measure of merit of the financial status of the company. Fundamental analysis is also affected by macroeconomic variables. These variables consist of gross domestic product, rate of unemployment, and rate of inflation. If the financial ratios are conditioned on macroeconomic variables, then the effect of macroeconomic variables on intrinsic values of securities can be investigated [7]. The economic state, whether the economy is expanding or contracting, also affects the investors’ analyses [8]. Overall, fundamental analysis numbers derive from firm-specific, industry-specific, and economy-related measures to determine the intrinsic value of a security. All three of these sources help examine the static values of stock prices.

In a fundamental analysis based on the financial ratios of the companies, a simple regression model for predicting the prices assuming four financial ratios DER, MBR, GPM, and QR can be written as follows:

$$P_{t+1} = \alpha + \beta_1 DER_t + \beta_2 MBR_t + \beta_3 GPM_t + \beta_4 QR_t + \theta_t \quad (1)$$

where  $P_{t+1}$  represents the firm’s share price at the next time;  $\alpha$ ,  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$ , and  $\beta_4$  are the regression parameters to be estimated; and  $\theta_t$  is an error term. The effect

of different financial ratios on the prices can be examined. More common financial ratios used in the literature for the purpose of fundamental analysis can be found in [9–11].

It should be noted that the financial ratios utilized in the fundamental analysis are collected from the annual statements of the companies. As it is shown by Devine in [12], the quarterly financial data may contain information inconsistent with those derived from the annual financial data, and the financial decision makers are advised to make their financial decisions based on annual financial ratios as compared with the quarterly reported financial ratios. Moreover, the annual financial statements report some financial ratios which may not be necessarily reported in a quarterly statement, including annual capital ratio factor and annual leverage.

## 1.2 Technical Analysis

Technical analysis is unlike fundamental analysis in predicting stock prices since it is based on the belief that stock prices already indicate information from financial statements. As a result, the intrinsic value of a security is the same as the market value. Technical analysis uses pattern recognition to project short-term stock prices from the previous data. Technicians study active changes in stock prices as well as chart trends to embrace short-term trading opportunities anywhere from a minute-by-minute basis to a weekly basis. They review past data since history is inclined to repeat itself [5]. Nonlinear patterns are extracted from noises to allocate the regularities and symmetries in the time series of the stock prices [4, 13]. Much of the research done on pattern recognition within stock prices looked at the analytical historical patterns in the calendar periods such as the result of the weekend, turn of the month, holiday, and first month of a new year [5]. Not only are historical prices studied, but also the volume of trades as the second market action. The historical trading volumes can assist in predicting the stock prices [14].

Even though technical analysis is as old as the stock market itself, it has been denounced due to its subjective nature. People have tried to compare visuals of technical analysis to that of human cognition by stating that pattern recognition is something computers do not have a mere privilege when compared to visual ability of a person [15]. So in return, technical analysis is beneficial in that it can assess the dynamics of stock prices, which is challenging to humans [15]. Technical analysis encapsulates the trading strategies created by practitioners from their own experiences in trading.

Predictability of stock prices has been subject of considerable debate among researchers and analysts in the past few decades [2, 16, 17]. However, this analysis is complex since stock market is dynamic and influenced by many factors. The stochastic process of stock prices is noisy, dynamic, nonlinear, nonparametric, nonstationary, and chaotic in their nature [18]. The characteristics of the stochastic process of stock prices as well as their definitions in mathematical terms are depicted in Table 1.

Moreover, the stochastic process of stock prices is nonstationary. Strong stationarity, weak stationarity, and non-stationarity are three forms of stationarity in

**Table 1** Complexity of the stochastic process of stock prices and definitions

Characterizations	Definitions
Nonparametric	Are not drawn from a given probability distribution with specified mean and variance
Noisy	Unexplained variability within the time series; corrupted; unstructured
Nonlinear	The change of the output is not proportional to the change of the input
Dynamic	Periodically updated; it changes asynchronously over time as new information becomes available
Chaotic	Highly sensitive so initial conditions

stochastic processes. Let  $\{X_t; t \in \mathbb{Z}\}$  be stochastic process. Assume the finite set of time indices  $\{t_1, t_2, \dots, t_s\} \subset \mathbb{Z}, s \in \mathbb{Z}^+$ , in which the time indices go from 1 to  $s$ , to set (1),  $\{X_t\}$ , as  $(X_{t_1}, X_{t_2}, \dots, X_{t_s})$ , and (2),  $\{X_{t+k}\}$ , as  $(X_{t_1+k}, X_{t_2+k}, \dots, X_{t_s+k})$ , where  $k \in \mathbb{Z}$ . The stochastic process  $X_t$  is strong stationary if the joint distribution of (1) and (2) are equal. The stochastic process  $X_t$  is weak stationary if  $E|X_t|^2 < \infty, \forall t, E(X_t) = \mu, \forall t$ , and  $\text{cov}(X_{t_1}, X_{t_2}) = \text{cov}(X_{t_1+k}, X_{t_2+k}), \forall t_1, t_2, k$ . However, the stochastic process of stock prices is nonstationary since the statistical properties of the time series of stock prices change over time. The Random Walk model has been tested in the literature for modeling the time series of stochastic process of stock prices [19] as follows:

$$X_t = \alpha + X_{t-1} + \varepsilon_t \quad (2)$$

where the error term  $\varepsilon_t$  is a white noise with mean 0 and variance  $\sigma^2, \varepsilon_t \sim \text{WN}(0, \sigma^2)$ . The statistical properties of the error term are  $E(\varepsilon_t) = 0, \text{var}(\varepsilon_t) = \sigma^2 < \infty$ , and  $\text{cov}(\varepsilon_t, \varepsilon_{t-k}) = 0, \forall t, k$ . It should be pointed out that, for the purpose of prediction, the volatility issue of financial time series as one of the characteristics of complexity needs to be taken into consideration. In a recent work, Mantalos et al. [20] study skewness as one of the causes of deviation from normality in financial time series. They propose a test procedure for skewness in autoregressive conditional volatility models which is applicable to financial datasets and capable of detecting the volatility of these datasets.

Neural network-based methods including Multilayer Perceptron have become popular techniques for forecasting stock movements due to their ability to learn how to do tasks through a learning algorithm [21]. Multilayer feedforward perceptron neural network (MLP) is appropriate for predicting stock prices and outperforms the statistical techniques such as regressions for the following four reasons. One, it is numeric and fits the financial datasets. Two, a distribution of input data is not required. Three, a model formulation is not required. Four, it can predict unseen data without reprocessing training data. However, determination of a MLP architecture including learning algorithms, the choice of number of layers, number of neurons in each layer, and (nonlinear) activation functions between the layers is complicated. Different network architectures result in different generalization errors; therefore, generalization error can be used as a measure to evaluate

the performance of the network. An appropriate network architecture can be achieved through hyper-parameter optimizations [22].

In this paper, we use a multilayer feedforward perceptron neural network (MLP) and predict the short-term stock trends based on fundamental and technical analysis in isolation. Next, we design a hybrid model using a nonlinear autoregressive with exogenous input (NARX) in neural networks by integrating the firm's financial ratios and historical stock prices and predict the future stock movements. The main goal of this study is to explore the predictability of short-term stock trends of Technology companies listed on NASDAQ based on fundamental and technical analysis using MLP as well as an approach for combining fundamental and technical analysis using a NARX in neural networks. It is shown that the hybrid model is beneficial for stock market participants, in the sense of producing return greater than a buy-and-hold (B&H) strategy.

This paper proceeds in seven sections. In Sect. 2, we conduct a brief review of related research. Section 3 introduces the multilayer feedforward perceptron neural network (MLP); Sect. 4 defines the input and output datasets as well as the data preprocessing methods; Sect. 5 proposes our fundamental, technical, and hybrid models; Sect. 6 presents the results and discusses the findings; Sect. 7 compares the proposed hybrid model with the state-of-the-art models; and Sect. 8 provides the conclusions to the work and offers future directions.

## 2 Literature Review

A voluminous amount of literature has examined the ability of soft computing techniques including neural networks [23–25], fuzzy systems [26], neuro-fuzzy systems [27], genetic algorithm [24, 28], particle swarm optimization [29, 30], etc. to predict the future stock movements as follows:

Roman and Jameel [25] use backpropagation and recurrent neural networks and predict stock returns in five stock markets including Canada, Hong Kong, Japan, UK, and USA. They measure the prediction accuracies as well as the stock returns for each market using two types of neural networks. They conclude that their proposed method for building portfolios across international stock markets successfully predicts the future stock market movements. Leigh et al. [24] test the conventional pattern recognition techniques, neural networks, genetic algorithms, and the cross validation for forecasting price changes for the New York Stock Exchange (NYSE) Composite Index. The results of their study reveal the effectiveness of the bull flag patterns and the volume pattern heuristic. Ang and Quek [31] examine the predictability of stock price differences based on stock trading rough set-based pseudo outer-product (RSPOP). Their trading model uses pseudo outer-product-based fuzzy neural network using the compositional rule of inference (POPFNN-CRI) and simple moving average trading rules. They evaluate the performance of their model against stock trading with dynamic evolving neural-fuzzy inference system (DENFIS), the stock trading without forecast model, and the stock trading with ideal forecast model. They find that RSPOP performs better than DENFIS and the stock trading without forecast model in terms of identifying the rules with greater interpretability and producing higher profits.

Chavarnakul and Enke [32] examine the profitability of stock trading using generalized regression neural networks (GRNN), volume-adjusted moving average (VAMA), and ease of movement (EMV) indicators by utilizing data from S&P 500 index. They use equivolume charts and show that their model outperforms the VAMA, EMV, and MA in isolation. In addition, they find that their model performs better than a B&H strategy. Atsalakis and Valavanis [27] use an adaptive neuro fuzzy inference system (ANFIS) and predict the short-term stock prices based on historical data from Athens and NYSE. They find that the proposed model outperforms the simple B&H strategy and 13 other existing models in terms of accuracy. Chang et al. [26] present a Takagi–Sugeno fuzzy rule-based model for trading stocks. Their model uses a set of trading indicators to generate trading signals learned by a support vector regression (SVR). Additionally, they compare the performance of their model against conventional linear regression model and artificial neural networks and report that their model yields more profit than other models. Boyacioglu and Avci [33] use an adaptive network-based fuzzy inference system (ANFIS) and predict the stock market return on Istanbul Stock Exchange (ISE). They use six macroeconomic variables and three indices as input data. Their experimental results indicate that their model successfully predicts the monthly return of ISE National 100 Index with an acceptable accuracy.

A summary of related work including the markets, methods, and the results is shown in Table 2.

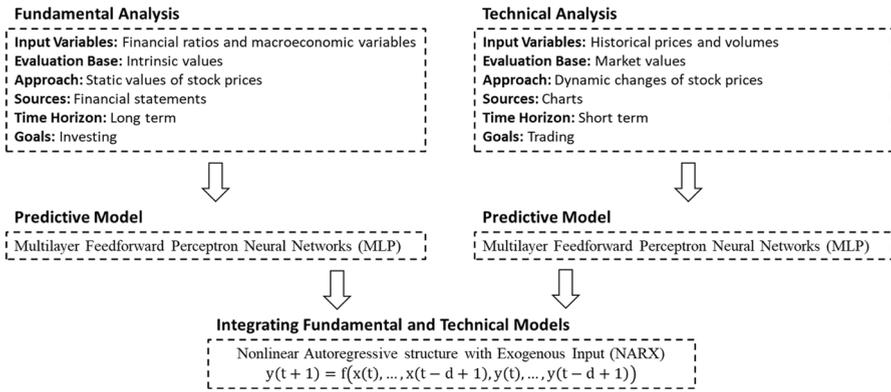
From the literature review, it can be observed that a variety of techniques have been used in both fundamental and technical analyses. In contrast to voluminous research on fundamental and technical analysis in isolation, there have been only very limited research aimed at developing an integrated approach based on both technical and fundamental analyses. An integrated model of fundamental and technical analysis using multilayer feedforward perceptron neural networks (MLP) and a nonlinear autoregressive structure with exogenous input (NARX), which is the subject of this study, has not previously been tested in the literature. Figure 1 demonstrates a comparison between fundamental and technical analyses as well as the schematic of our approach to predicting short-term stock trends using our neural networks.

### 3 Multilayer Feedforward Perceptron Neural Networks (MLP)

Multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN) and uses backpropagation as a technique for training. In multilayer perceptron neural networks, the output of each layer forms the input of the next layer. In this study, we use a three-hidden-layer MLP in order to obtain the optimum results.  $P.H_1.H_2.H_3.Q$  describes the architecture of an MLP with three hidden layers, where  $H_1, H_2,$  and  $H_3$  are the hidden layers, and  $P$  and  $Q$  are the input and output layers, respectively. This network can be translated into a matrix form as a function:  $R^D \rightarrow R^L$ , where  $D$  is the size of input vector  $x$ , and  $L$  is the size of the output vector  $f(x)$  [36].

**Table 2** An overview of different approaches to stock market price prediction

References	Markets	Methods	Results
[25]	Canada, Hong Kong, Japan, UK, and USA	Backpropagation and recurrent neural networks	The proposed method successfully predicts the future stock markets movements
[24]	NYSE Composite Index	Pattern recognition techniques, neural networks, genetic algorithms and cross validation	The results of their study reveal the effectiveness of the bull flag patterns and the volume pattern heuristic
[31]	Artificial Price Data	Rough set-based pseudo outer-product (RSPOP)	RSPOP performs better than Dynamic Evolving Neural-Fuzzy Inference System DENFIS
[32]	S&P 500 Composite Index	Generalized regression neural networks (GRNN)	Their model performs better than a B&H strategy
[27]	Athens Stock Exchange and New York Stock Exchange (NYSE)	Adaptive neuro fuzzy inference system (ANFIS)	The proposed model outperforms the simple B&H strategy and 13 other existing models in terms of accuracy
[26]	Apple Inc., Boe-ing Company, Caterpillar Inc., Johnson & Johnson, Exxon Mobil Corporation, Verizon Communications Inc. and S&P 500	Takagi-Sugeno fuzzy rule-based model	Their model yields more profit than Support Vector Regression and Artificial Neural Networks models
[33]	Istanbul Stock Exchange	Adaptive network-based fuzzy inference system (ANFIS)	Their model successfully predicts the monthly returns of ISE National 100 Index with an acceptable accuracy
[34]	S&P 500 Index	Long short-term memory (LSTM)	Their proposed LSTM model outperforms the previous models in terms of Daily Returns, Sharpe Ratio, and Accuracy
[35]	S&P 500 Index, Individual Companies	Hierarchical graph attention network	The proposed model selects information automatically and outperforms the existing models



**Fig. 1** Comparison between fundamental and technical analyses as well as the schematic of the proposed approach to predicting short-term stock trends

Assuming a training dataset  $D = \{x_n, t_n\}_{n=1}^N$ , we can form a data matrix  $X$  containing the training data as follows:

$$X = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix} \quad (3)$$

$N \times (P + 1)$

In order for the MLP to learn the relationships between the variables based on the training data, the first stage is to build the product that connects the input layer to the first hidden layer:

$$y_1 = \Omega_1 x \quad (4)$$

$H_1 \times (P + 1) = H_1 \times (P + 1) \times (P + 1) \times 1$

where  $\Omega_1$  is the first weight matrix. Similarly, the first and second hidden layers, the second and third hidden layers, and the last hidden layer and the output layer are connected as follows:

$$y_2 = \Omega_2 y_1 \quad (5)$$

$H_2 \times (H_1 + 1) = H_2 \times (H_1 + 1) \times H_1 \times (P + 1)$

$$y_3 = \Omega_3 y_2 \quad (6)$$

$H_3 \times (H_2 + 1) = H_3 \times (H_2 + 1) \times H_2 \times (H_1 + 1)$

$$z = \gamma y_3 \quad (7)$$

$Q \times 1 = Q \times (H_3 + 1) \times H_3 \times (H_2 + 1)$

where  $\Omega_2$ ,  $\Omega_3$ , and  $\gamma$  are the second, third, and the last weight matrices, respectively. The dimension of each matrix/vector in Eqs. (4) to (5) varies depending on the

number of neurons in the input, hidden, and output layers of MLP. For instance, Eq. (4) explains the relationship between the input layer and the first hidden layer; therefore, the dimension of the corresponding matrix/vector is the number of neurons in the product between the input layer and the first hidden layer,  $P \times H_1$ . The dimension of the other matrices/vectors is determined in a similar manner. Therefore, the MLP fitted model can be defined as follows:

$$mlp(x) = f_Q(z) = f_Q(\gamma[1, f_{H_3}(\Omega_3[1, f_{H_2}(\Omega_2[1, \{f_{H_1}(\Omega_1 x)\}^\tau]^\tau)^\tau]^\tau]^\tau) \tag{8}$$

MLP’s neurons use nonlinear activation functions. The MLP’s two common activation functions are both sigmoids: Tan-Sigmoid activation function (tansig) and Log-Sigmoid activation function (logsig). Tan-Sigmoid activation function is as follows:

$$tansig(n) = 2/(1 + \exp(-2 * n)) - 1 \tag{9}$$

which ranges between  $-1$  and  $1$ . While, Log-Sigmoid activation function is as follows:

$$logsig(n) = 1/(1 + \exp(-n)) \tag{10}$$

which ranges between  $0$  and  $1$ .

The MLP is made up of three or more layers, including an input, output, and one or more hidden layers. Every neuron in one layer attaches with a specific weight,  $w_{ij}$ , to another neuron in the next layer. After each data piece is processed, that is when the learning occurs in the perceptron from altering connection weights depending on the amount of error in the output. This error amount is determined by comparing it to the anticipated result. The error in the output neuron  $j$  in the  $n$ th data point is calculated by

$$e_j(n) = d_j(n) - y_j(n) \tag{11}$$

where  $d$  is the target value and  $y$  is the value produced by the perceptron. The weights are adjusted according to the corrections that minimize the error in the entire output, given by

$$\mathcal{E}(n) = \frac{1}{2} \sum_j e_j^2(n) \tag{12}$$

Using gradient descent, the change in each weight is

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n) \tag{13}$$

where  $y_i$  is the output of the previous neuron, and  $\eta$  is the learning rate, which is selected in order to ensure that the weights converge quickly to a response, without any oscillation.

The derivative calculation depends on the induced local field  $v_j$ , which itself varies. It can be proved that for an output neuron, this derivative can be simplified to

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n) \phi'(v_j(n)) \quad (14)$$

where  $\phi'$  is the derivative of the activation function as described previously. The analysis is more complicated for the change in weights with respect to a hidden neuron, but it can be shown that the relevant derivative is

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n)) \sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} w_{kj}(n) \quad (15)$$

This depends on the change in weights of the  $k$  th neurons, which represent the output layer. Therefore, to change the hidden layer weights, the output layer weights change based on the derivative of the activation function.

The performance of the multilayer perceptron can be evaluated based on two measures: accuracy and generalization error. Accuracy represents the learning ability of the model. Generalization error shows how accurately the model is capable of predicting unseen data. Generalization error is an appropriate indicator of performance of neural networks. Hyper-parameter optimization is the process of selecting weights from the architecture in order to improve the performance of the learning algorithms and obtaining the optimal solutions. Hyper-parameter optimizations can be used in order to achieve the best network architecture that results in the least generalization errors. In this study, we use MLP hyper-parameter optimizations to obtain the best network architecture with respect to generalization error and accuracy.

## 4 Data Preprocessing Methods

Finance, energy, healthcare, technology, etc. are popular sectors of stock markets. In this paper, in order to lower the noise level, we aim to study firms that belong to the same sector. Hence, we chose 578 Technology companies whose data is available on NASDAQ website. Financial ratios of the companies are the main input variables in fundamental analysis of the stock market. This study focuses on twenty-four financial ratios compiled from the financial statements of the Technology companies in the fiscal year of 2017, selected on the basis of the previous literature and for reasons of importance and tractability.

### 4.1 Feature Selection

Twenty-four financial ratios for the fundamental analysis of stock movements are studied in this subsection. These twenty-four financial ratios are selected for the reason of significance in terms of their effect on the stock movements. These financial ratios are further used as input variables in the multilayer feedforward perceptron model for predicting the stock movements. These predictors are listed as  $\{fr_n\}_{n=1}^{24}$ , where  $fr$  is the financial ratio and  $n$  is an index for the financial ratios ranging

between 1 and 24. The set of financial ratios as predictors in the multilayer feedforward perceptron model is shown below:

$$\{\text{fr}_n\}_{n=1}^{24} = \{\text{fr}_1, \text{fr}_2, \text{fr}_3, \dots, \text{fr}_{24}\} \quad (16)$$

#### 4.1.1 Determination of the Optimal Feature Set

The approach to the determination of the optimal set of features used in this study attempts to minimize the mean squared errors (MSEs) between the actual and predicted values as shown below:

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2 \quad (17)$$

where  $y_t$  is the actual value,  $\hat{y}_t$  is the predicted value,  $t$  is time, and  $N$  is the total number of days for computing MSEs. The multilayer feedforward perceptron models are built for various combinations of the financial ratios as the predictors, and the resulting MSEs are computed. The process of determination of the optimal feature set stops when the following condition is satisfied:

$$\text{The subset of the features is 'PURE' i.e., } \text{MSE}_1 \leq \text{Err}_{\text{tol}} \cdot \text{MSE}_{\text{total}} \quad (18)$$

where  $\text{MSE}_1$  indicates the MSE of the predictive model built based on the corresponding subset of features,  $\text{MSE}_{\text{total}}$  indicates the MSE of the predictive model built based on the original set of twenty-four financial ratios, and  $\text{Err}_{\text{tol}}$  is a predefined acceptable tolerance (i.e.,  $\text{Err}_{\text{tol}} = 0.8$  in this study).

Twelve financial ratios are selected as the most significant variables through the Feature Selection preprocess [37] by conducting the hyper-parameter optimizations. The selected financial ratios are as follows: current ratio ( $\text{fr}_1$ ), inventory/current assets ( $\text{fr}_3$ ), inventory day sales ( $\text{fr}_5$ ), net income/common equity ( $\text{fr}_6$ ), net income/net sales ( $\text{fr}_7$ ), net income/total assets ( $\text{fr}_{11}$ ), net sales/property-plant and equipment ( $\text{fr}_{12}$ ), net sales/work capitalization ( $\text{fr}_{17}$ ), pretax income/net sales ( $\text{fr}_{19}$ ), quick ratio ( $\text{fr}_{21}$ ), total liability/common equity ( $\text{fr}_{23}$ ), and total liability/total assets ( $\text{fr}_{24}$ ), as reported in Table 3.

#### 4.2 Data Discretization

Next, the financial ratios, which are continuous variables, are discretized into discrete counterparts through a data discretization preprocess by conducting the hyper-parameter optimizations. The twelve financial ratios are discretized as shown in Table 3. For instance, inventory/current assets are discretized into the following five classes: 0.1, 0.3, 0.5, 0.7, and 0.9. The combination of twelve financial ratios discretized into their classes, contributes to the least mean squared error (MSE) by comparing the simulated results from MLP neural networks and actual data.

**Table 3** List of financial ratios, variables, and discretization numbers determined through feature selection and data discretization preprocesses

Financial ratios	Variables	Discretization numbers
Current ratio	fr <sub>1</sub>	2
Inventory/current assets	fr <sub>3</sub>	5
Inventory day sales	fr <sub>5</sub>	4
Net income/common equity	fr <sub>6</sub>	2
Net income/net sales	fr <sub>7</sub>	6
Net income/total assets	fr <sub>11</sub>	3
Net sales/property-plant and equipment	fr <sub>12</sub>	6
Net sales/work capitalization	fr <sub>17</sub>	1
Pretax income/net sales	fr <sub>19</sub>	4
Quick ratio	fr <sub>21</sub>	4
Total liability/common equity	fr <sub>23</sub>	6
Total liability/total assets	fr <sub>24</sub>	4

In our technical analysis, we study stock prices of the same companies to forecast the future stock movements. We normalize the input data to reduce the data redundancies and improve the data integrity. The number of input data may vary depending on how much of an impact the historical stock prices have on the future prices [38]. We use IXIC daily stock prices from 2013–2003 to 2018–2006. In order to enhance the comparison between the stock prices, we should restructure the time series of stock prices to achieve stationarity. First, we convert the stock prices into a logarithmic format since stock prices are based on returns, and returns are based on percentages. Next, we calculate the differences between each time series point and multiply them by 100. This computes the compound returns,  $r$ . Time series of adjusted closing prices has a moving average while time series of compound returns has a constant mean, variance, and autocorrelation. This indicates that our data is now stationary.

In addition, we determine the autocorrelations in order to further examine the time series of compound returns. The autocorrelation between any two data points  $r_t$  and  $r_{t-h}$  only depends on the time lag  $h$  between them. Lag- $h$  covariance is defined as:

$$\gamma_h = \text{cov}(r_t, r_{t-h}) \quad (19)$$

Therefore, the theoretical lag- $h$  autocorrelation is

$$\rho_h = \text{corr}(r_t, r_{t-h}) = \frac{\gamma_h}{\gamma_0} \quad (20)$$

where  $\gamma_0$  is the lag-0 covariance.

The theoretical partial autocorrelation  $\phi_{h,h}$  is the autocorrelation between  $r_t$  and  $r_{t-h}$  after removing the effect of confounders. The theoretical autocorrelation and partial autocorrelation can be estimated using sample autocorrelation and sample

partial autocorrelation. For the data points  $r_1, r_2, \dots, r_T$ , the sample lag- $h$  autocorrelation is:

$$\hat{\rho}_h = \frac{\sum_{t=h+1}^T (r_t - \bar{r})(r_{t-h} - \bar{r})}{\sum_{t=1}^T (r_t - \bar{r})^2} \tag{21}$$

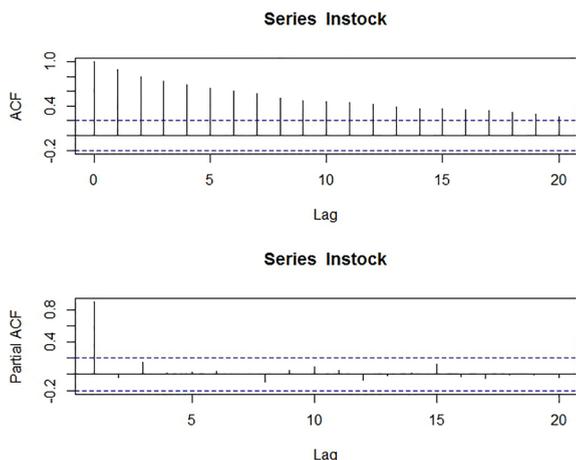
where  $\bar{r}$  is the sample mean. Similarly,  $\hat{\phi}_{h,h}$  denotes the sample partial autocorrelation. The autocorrelation function (ACF) and partial autocorrelation function (PACF) are illustrated in Fig. 2. It is noted that PACF is the sequence  $\phi_{h,h}$ ,  $h = 1, 2, \dots, N - 1$  for the time series  $r_t$ ,  $t = 1, 2, \dots, N$  after removing the linear dependence on  $r_1, r_2, \dots, r_{t-h+1}$ .

Each vertical line, in both graphs in Fig. 2, represents the correlation between the lags. It can be seen in the upper graph that there is a very gradual descent in the lags in the ACF depicted in the upper graph. However, there is an immediate drop in the correlation of the first lags in the PACF depicted in the lower graph. From the autocorrelations demonstrated in this correlogram, we can infer that our data is stationary.

### 4.3 Self-Organizing Maps

Self-organizing maps (SOM) supply a data visualization method that, in return, assists in understanding high-dimensional data by condensing the dimensions of data into a map. This method groups together similar data which is a clustering concept. As a result, SOM not only shows similarities between data, but also decreases data dimensions. SOM algorithm linearly or randomly assigns the values of weight vectors of each output. Then, Euclidean distance can be used to gauge weight vectors of all the neurons and the data points represented. A neuron that is the closest distance wins and is moved closer to the data point. The SOM

**Fig. 2** Correlogram of sample autocorrelation function (ACF) and partial autocorrelation function (PACF)



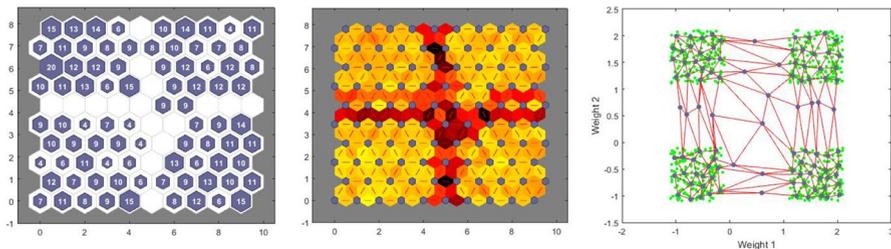
shows information that is learned. Therefore, SOM reflects an input value onto a point in the place and the surrounding points have alike functions [39, 40]. SOM is used to associate the historical stock price data points and illustrate the results.

In the technical analysis, the input to the SOM can be written as  $I_{\text{SOM}} = \{X_{\text{Disc}}, Y_{\text{Obs}}\}$ , where  $X_{\text{Disc}}$  is the series of discretized past stock prices,  $Y_{\text{Obs}}$  is the observed stock price. In this paper, as the cluster configuration of the  $a \times b$  and the optimal quantity of clusters are unknown, the training data is used to generate various cluster combinations. The localization is performed by modifying the learning rate  $\alpha(t)$  and the neighborhood size  $R(t)$ . The learning amount for each neuron can be computed by  $\alpha(t) * e^{-d/R(t)}$ , where the distance  $d = \|y - w\|$  given  $y$  as the input and  $w$  as the corresponding weight. It should be noted that as  $\alpha(t)$  and  $R(t)$  decrease with time, the learning amount will decrease as well, and after sufficient iterations, the updating will reach saturation. These results are shown in Fig. 3.

Hits plot displays the neuron location in the topology along with the training data correlated with each of the neurons. There are 100 neurons due to the 10–10 grid. It can be seen that the maximum number of hits any individual neuron has is 20, meaning there are 20 input vectors in the given cluster. In SOM neighbor weight distances plot, the blue hexagons represent the neurons. Meanwhile, the red lines connect the neighboring neurons. The colors within the red line region show the distances between the neurons. Larger distances are shown by darker colors while smaller distances are shown in lighter colors. There is a dark segment band from the lower-center region into the upper-right region. This shows the SOM network grouped the discretized representation of the input space into four apparent groups. In SOM weight positions plot, the location of the data points and weight vectors are displayed. As shown above, the map is well dispersed throughout the input space. The determination of optimal cluster combination from the possible configurations is discussed in the following subsection.

### 4.3.1 Determination of the Optimal Number of SOM Clusters

In this paper, the optimal number of SOM clusters is determined through the Silhouettes methodology introduced in [41]. Assume the SOM configuration  $a \times b$ , the quantity of clusters for this configuration can be calculated as  $k = a + b$ , and the



**Fig. 3** The results of the self-organizing map for clustering the input data in technical analysis: hits (left), SOM neighbor weight distances (middle), and SOM weight positions (right)

Silhouettes  $S_j(i)$  for the  $i$  th point in the  $j$  th cluster  $C_j$ , denoted by  $p_j(i)$  can be written as:

$$S_{C_j}(i) = \frac{b_j(i) - a_j(i)}{\max\{a_j(i), b_j(i)\}} \quad (22)$$

where

$$a_j(i) = \frac{1}{|C_j| - 1} \sum_{h \neq i, h \in C_j} \|p_j(i) - p_j(h)\| \quad (23)$$

and

$$b_j(i) = \min_{C_i} \left\{ \sum_{h=1}^{|C_i|-1} \|p_j(i) - p_C(h)\| \right\}, \quad i = 1, 2, \dots, k \quad (24)$$

Larger value of  $S_{C_j}(i)$  means the point  $p_j(i)$  is well clustered and that point belongs to the cluster  $C_j$ . Next, we calculate the average Silhouettes value for each cluster  $C_i$  in all the configurations as:

$$\mu_{C_i} = \frac{1}{C_i} \sum_{j=1}^{\|C_i\|} S_{C_i}(j), \quad i = 1, 2, \dots, L \quad (25)$$

where  $L$  is sum of all the clusters of all the combinations. Once the Silhouettes values for all the points in all the clusters are determined, the median Silhouettes value is determined as

$$\text{median}_{\mu_C} = \text{median}\{\mu_{C_1}, \mu_{C_2}, \dots, \mu_{C_L}\} \quad (26)$$

Then, the SOM cluster combination with all the clusters having  $\mu_{C_i} \geq \text{median}_{\mu_C}$  will be the optimal cluster combination.

## 5 Model Development

In this section, two separate models are presented in order to study the predictability of stock movements using the fundamental and technical analysis in isolation. Then, a hybrid model is examined and compared to the first two models. The number of layers, number of neurons in each layer, and the (nonlinear) activation functions between the layers are determined through MLP hyper-parameter optimization. The results of the optimized network reveal that the (nonlinear) activation functions between the input layer and the 1st hidden layer, as well as the (nonlinear) activation function between the 3rd hidden layer and the output layer, are Log-sigmoid. Interestingly, the same type of (nonlinear) activation function connects the hidden layers. There are 60, 51, and 1 neurons in the 1st, 2nd, and 3rd hidden layers of the optimized network, respectively. These numbers are selected based on the optimum

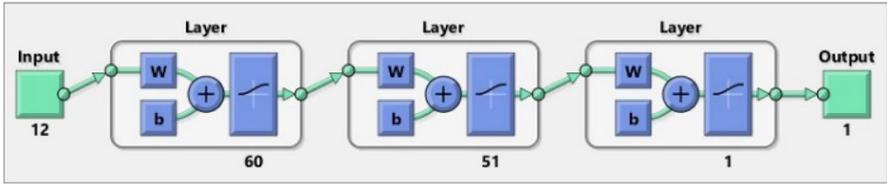


Fig. 4 Multilayer perceptron (MLP) network implemented for the prediction of stock through fundamental analysis

results after hyper-parameter optimization. The neural network depicted in Fig. 4 represents the fundamental analysis of the stock market. According to the results of the fundamental analysis, this model contributes to the least MSE among all the trials of the hyper-parameter optimization process.

Similar to the fundamental analysis, we conduct MLP hyper-parameter optimizations for the technical analysis to obtain the optimum results. There are 30, 32, and 1 neurons in the 1st, 2nd, and 3rd hidden layers, respectively. The (nonlinear) activation functions between the input layer and the 1st hidden layer, the 1st and the 2nd hidden layers, and the 2nd and the 3rd hidden layers are all Log-sigmoid. However, the last hidden layer, which contains only one neuron, is connected to the output layer via a Tan-sigmoid activation function. The optimum results indicate that this (nonlinear) activation function results in smaller MSEs as compared to Log-Sigmoid. Figure 5 displays the architecture of the neural network in the technical analysis, which results in the least MSE among all the trials.

Next, we present a hybrid model by combining the fundamental and technical analyses. We combine the models in order to integrate financial ratios and historical prices, and then we see how this affects the simulated results. Figure 6 illustrates the prediction system used in order to predict the future stock movements. The inputs of the hybrid model are the same financial ratios and stock prices examined by the first two models, and the output of this model is the same as the output of those models.

The three-hidden layer MLP neural network learns the relationship between the stock prices, financial ratios, and the future stock movements. The hybrid model computes a predicted value  $y(t + 1)$  as depicted below, given past values of  $y(t)$ , which represent the stock price time series, and another input dataset  $x(t)$ , which are the financial ratios of the companies selected using a feature selection preprocess:

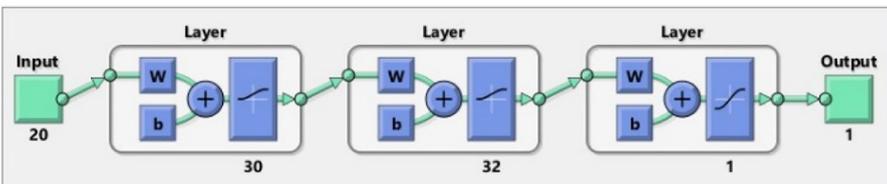
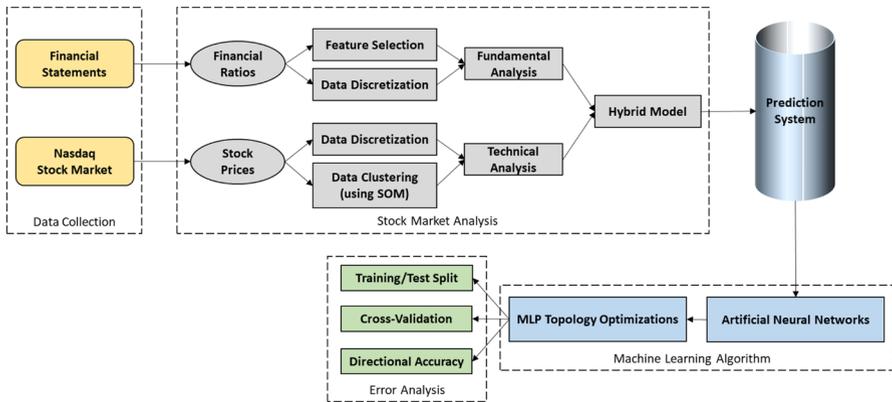


Fig. 5 Multilayer perceptron (MLP) network implemented for the prediction of stock prices through technical analysis

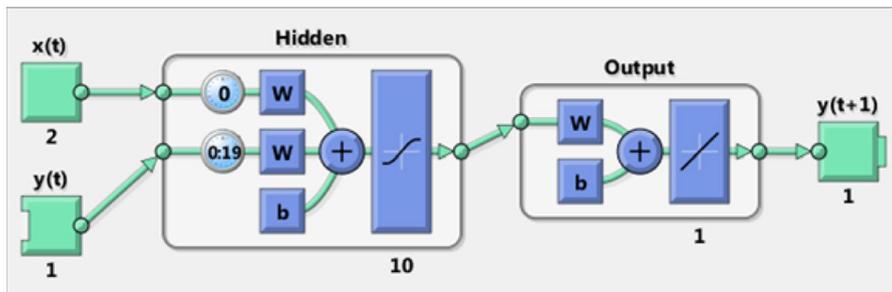


**Fig. 6** Prediction system of short-term stock trends based on the hybrid model of fundamental and technical analysis

$$y(t + 1) = f(x(t), \dots, x(t - d + 1), y(t), \dots, y(t - d + 1)) \tag{27}$$

In addition to the past values of the same series and the driving (exogenous) variable, the model contains an error term. The error term indicates that information about the other terms does not enable the future value of the time series to be predicted exactly. The architecture of the neural network representing the hybrid model of the fundamental and technical analysis is illustrated in Fig. 7.

In order to predict one-step-ahead stock price,  $y(1)$ , the hybrid model uses 12 financial ratios from the previous year financial statements as well as 20 past prices as input variables. More specifically, the number 0 in the hidden layer of the NARX model relates to  $x(0)$  which indicates that we use financial ratios from the previous year in our hybrid model. In a similar manner, the range 0:19 in the hidden layer of the NARX model relates to the series of variables  $y(0), y(-1), \dots, y(-19)$  which indicates that we use 20 past stock prices in our hybrid model.



**Fig. 7** Nonlinear autoregressive with external (exogenous) input (NARX) implemented for the prediction of stock prices through hybrid model

## 6 Results and Findings

In this section, first, we evaluate the performance of our models through a training/test split and cross-validation test. Next, we examine the directional accuracies of our simulated results using the three different models.

### 6.1 Training/Test Split and Cross-Validation Tests

In order to validate the fundamental, technical, and hybrid models, we test the accuracy and generalization error of each model. We use MSE as the measure of risk function and compare the simulated results with actual data obtained from NASDAQ. First, we split our data into two subsets: training and test data. The process of training a neural network is a very complex task which may have a significant effect on the ultimate results. We use 80% of our normalized data, which includes historical stock prices from 1992 Jan to 2013 Feb, for training the neural networks. The specifications of the training and testing datasets are shown in Table 4.

MSE of the model on the training dataset captures the accuracy of the model. On the other hand, MSE of the model on the test dataset represents the generalization errors. From the MSE on the training and test datasets for the fundamental, technical, and hybrid models, the following items are readily noticeable:

1. Fundamental analysis has the best generalization ability and results in the worst accuracy.
2. Technical analysis results in the best accuracy and has the worst generalization ability.
3. MSE of the hybrid model on training data is 0.0058 which indicates the generalization ability of the hybrid model.
4. The hybrid model contributes to the best results due to its overall performance on training and test datasets.

Furthermore, we split our data into three subsets: training, validation, and test data, and then we use cross-validation in order to prevent overfitting and underfitting in the hybrid model. Overfitting and underfitting affect the predictability of the hybrid model. If overfitting happens, the hybrid model learns the noises instead of the trends in the data. On the other hand, underfitting happens when the model misses the relationships between the input and output variables. The relation between objectives to be achieved and the estimated stock prices are shown in Fig. 8. The results are shown for all the data used as well as the training, validation, and test datasets, separately. These plots demonstrate the networks' outputs with respect to the targets for training, testing, and validation datasets. For a perfect fit, the correlation coefficients, R values, are supposed

**Table 4** Training and testing datasets

Sets	From	To	Count
Training	1992 Jan	2013 Feb	5480
Testing	2013 Mar	2018 Jun	1360

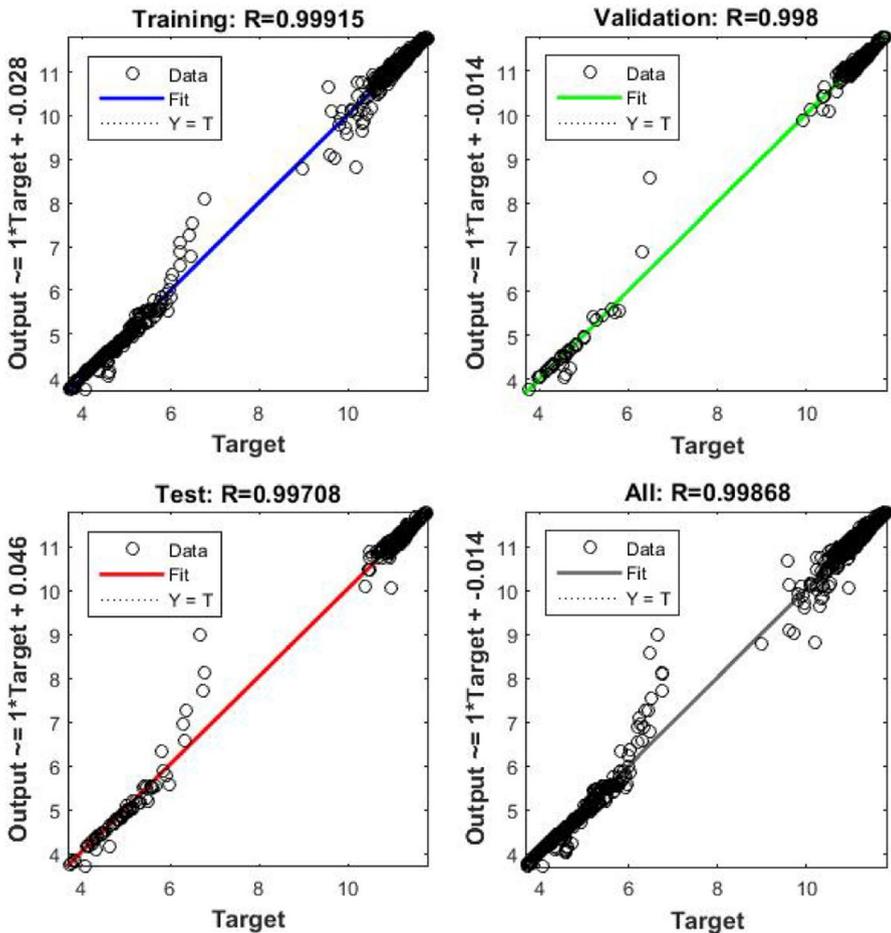


Fig. 8 The hybrid model performance for the training, validation, test, and all datasets

to be equal to zero meaning that outputs = targets. This can be achieved only if the data falls along a 45-degree line.

As it can be seen in Fig. 8, the correlation coefficients for the training, validation, test, and all datasets are 0.99915, 0.998, 0.99708, and 0.99868, respectively. It is observed that the cross-validation test verifies the quality of the hybrid model being able to accurately reproduce the stock prices. If more accurate results were required, the model could be retrained to modify the initial weights and biases of the networks and obtain an improved neural network. The error histogram with 20 bins is also examined and shown in Fig. 9 to identify the outliers and obtain additional verification of the neural network performance. The blue, green, and red bars represent the training, validation, and test datasets, respectively. From Fig. 9, the MSEs are

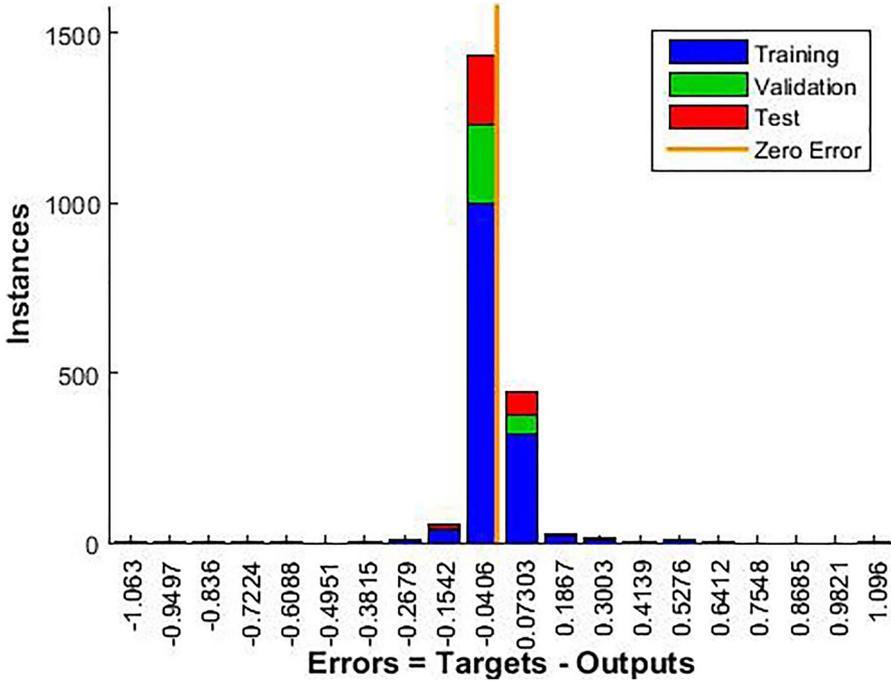


Fig. 9 Error histogram with 20 bins for the hybrid model

distributed within a reasonably good range around zero. It can be understood that the data fitting of the hybrid model is quite precise.

The time-series response of the model is illustrated in Fig. 10. From Fig. 10, it can be seen that the hybrid model predicts the future short-term trends successfully. Time-series response plot displays the inputs, targets, and errors versus time. It also indicates which time points were selected for training, testing, and validation.

### 6.2 Directional Accuracy

In order to examine our hybrid model in terms of directional accuracy of the simulated results, we predict the future stock returns and compare them with actual data obtained from NASDAQ. The outputs of both fundamental and technical analysis are discretized into five classes similar to the classes of input data in the technical analysis. The discretization numbers 1, 2, 3, 4, and 5 represent a sharp fall, downward trend, non-trending, upward trend, and sharp rise in stock returns, respectively. The numerical results for the directional accuracies are shown in Fig. 11.

In all cases, directional accuracies obtained from fundamental analysis are better than those reported by the technical analysis. The reason is the ability of MLP for fundamental analysis to generalize the results. In all cases, the hybrid model performs better than the other two models in terms of directional accuracies with an exception

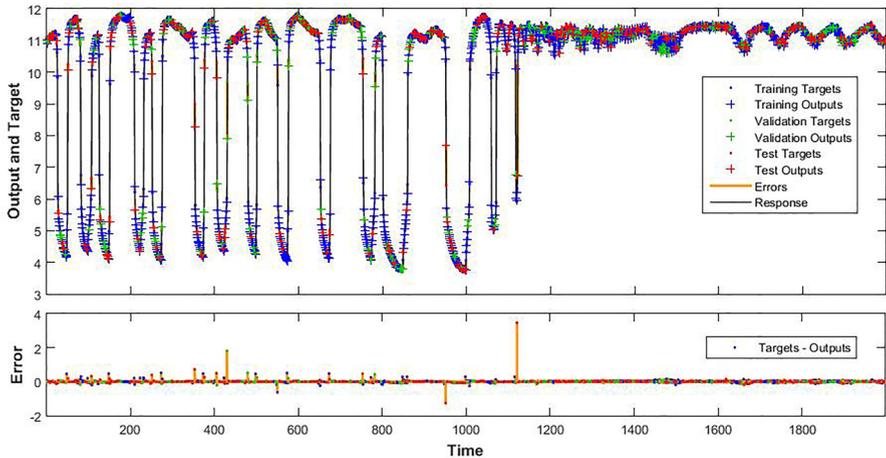


Fig. 10 Time-series response of the hybrid model

of stocks with downward trend. The average directional accuracy obtained from the hybrid model is 70.36% that is sizable relative to results of a B&H strategy. These results are consistent with the previous study by Bettman et al. [9] who showed that fundamental and technical analyses are complements rather than substitutes. Moreover, the MLP model presented in this paper can be applied to many stochastic processes with uncertainties including financial time series and time series of processes in physics, physiology, and biology. Our proposed model is capable of learning the prediction of one time series given the historical values of the same time series, the feedback, and another time series known as the external or exogenous time series.

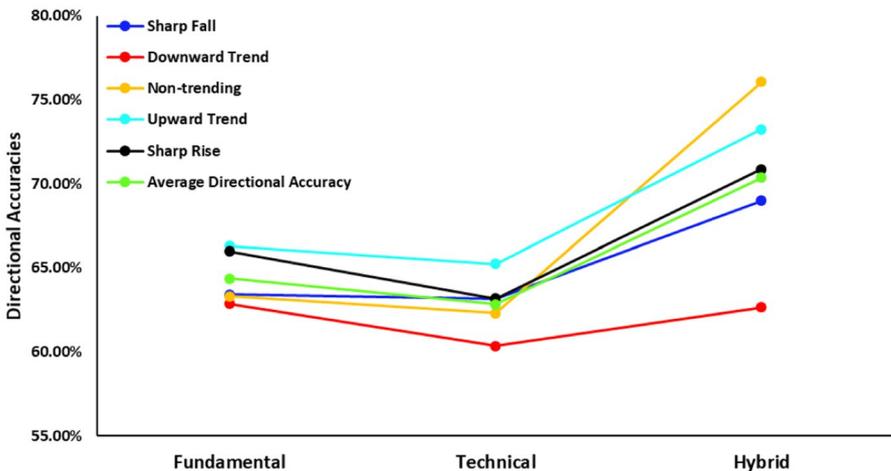


Fig. 11 Directional accuracies of the fundamental, technical, and hybrid models categorized into sharp fall, downward trend, non-trending, upward trend, and sharp rise

## 7 Comparisons with State-of-the-Art Models

The state-of-the-art predictive models are presented in the following subsections.

### 7.1 Regression, ANN, SVM

In this subsection, we illustrate comparisons between the correlation coefficients resulting from our hybrid model and three other models: Regression model [21], ANN [21, 42], and support vector machines (SVM) [21, 43].

A multivariate linear regression model for predicting stock returns as in [21] can be written as:

$$\hat{y} = a_0 + \sum_{i=1}^{27} a_i x_i \tag{28}$$

where  $y$  is the stock return and  $x_i$  is the  $i$  th independent variable, including the return with 1 day lag, the return with 2 days lag, etc. The model parameters  $a_0$  and  $a_i$  are estimated through least square estimation. The predicted stock return  $\hat{y}$  is developed as follows:

$$\begin{aligned} \hat{y} = & 6.9312 - 0.0234 * x_1 + 0.13 * x_2 + 0.021 * x_3 + 0.021 * x_4 \\ & - 0.021 * x_5 - 10.303 * x_6 + 6.0031 * x_7 + 0.7738 * x_8 \\ & + 0.2779 * x_9 - 0.43916x_{10} - 0.27754 * x_{11} + 0.12733 * x_{12} \\ & - 0.058638 * x_{13} + 13.646 * x_{14} + 9.5224 * x_{15} - 0.0003 * x_{16} \\ & + 0.24856 * x_{17} - 0.0016 * x_{18} + 0 * x_{19} - 2.334 \times 10^{-9} * x_{20} \\ & + 0.16257 * x_{21} + 0.63767 * x_{22} - 0.14301 * x_{23} \\ & + 0.08 * x_{24} + 0.074 * x_{25} - 0.0002 * x_{26} + 0.026301 * x_{27} \end{aligned} \tag{29}$$

The neural network model with sigmoid function with 27 input nodes and one output node which represents the stock return as in [21] will have the model setting and parameters as presented in Table 5.

In the support vector machine (SVM) as in [21], the function  $f(x)$  which estimates the stock return in terms of the dot products between the data is written as:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(x_i, x) + b \tag{30}$$

**Table 5** The settings and parameters of the neural network model

Maximum number of epochs	500
Number of hidden layers	1
Number of neurons in hidden layer	20
Learning rate	0.5
Momentum	0.2

**Table 6** The common kernel functions in SVM

Gaussian kernel function (radial basis functions—RBF)	$k(x_i, x_j) = \exp - \frac{\gamma \ x_i - x_j\ ^2}{2\sigma^2}$
Hyperbolic tangent kernel function (Sigmoid)	$k(x_i, x_j) = \tanh(\gamma \cdot x_i \cdot x_j + \beta)$
Polynomial kernel function	$k(x_i, x_j) = (\gamma \cdot x_i \cdot x_j + \beta)^m$

where  $\alpha_i$  and  $\alpha_i^*$  are the dual variables,  $b$  is bias, and  $k$  is the kernel function. We can write the objective functions for the SVM algorithm in terms of the dot products between vectors as follows:

$$\max - \frac{1}{2} \sum_{i=1}^1 \sum_{j=1}^1 (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) k(x_i, x_j) - \varepsilon \sum_{i=1}^1 (\alpha_i + \alpha_i^*) + \sum_{i=1}^1 y_i (\alpha_i - \alpha_i^*) \tag{31}$$

$$\text{s.t. } \sum_{i=1}^1 (\alpha_i - \alpha_i^*) = 0 \tag{32}$$

where the kernel function is the dot product between the feature maps,  $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ . The common kernel functions (Gaussian, polynomial, and hyperbolic tangent), are shown in Table 6.

In this SVM model, RBF is used as the kernel function. The parameters of the kernel function as well as the other SVM model parameters and the dual variables are estimated for prediction of each stock return and then the model is updated by adding the next stock return to the existing data. This process is repeated until the last stock return is predicted by using SVM.

The results of the training and testing datasets are displayed in Table 7.

From Table 7, it can be seen that the proposed hybrid model outperforms the regression, ANN, and SVM models in terms of the correlation coefficients on the testing datasets since it yields the greatest value. In terms of the correlation coefficients on the training datasets, our proposed hybrid model results in a greater value as compared to the regression and ANN models. However, the proposed hybrid model and the SVM model presented by Sheta et al. [21] have similar performance in terms of the correlation coefficients on the training datasets as there is a relatively small difference between their  $R$  values (0.99915 and 0.9995, respectively).

**Table 7** A comparison between the correlation coefficients of the proposed hybrid model, regression, ANN, and SVM

Datasets	Proposed model		Regression model		ANN model		SVM model	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Correlation coefficients	0.99915	0.99708	0.998	0.995	0.999	0.990	0.9995	0.9928

### 7.2 Long Short-Term Memory

Long short-term memory (LSTM), which belongs to the recurrent neural networks, is designed for learning long-term dependencies and has the capability of capturing the exploding gradients in financial data, as shown in [34]. In LSTM, the primary characteristic is contained in the hidden layer with units known as memory cells. Each memory cell includes three gates: a forget gate  $f_t$ , an input gate  $i_t$ , and an output gate  $o_t$ . The schematic of a memory cell with three gates is depicted in Fig. 12.

As it can be seen from Fig. 12, each gate contains an element of the input sequence at the current time  $x_t$  and the output at the previous time  $h_{t-1}$ . In this LSTM, the forget gate specifies which information is removed from the cell state as written below:

$$f_t = \text{sigmoid}(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \tag{33}$$

Next, the input gate specifies which information is added to the cell state, which includes the operations for the computation of the candidate values  $\tilde{s}_t$  and the computation of the activation values  $i_t$  as written below:

$$\tilde{s}_t = \tanh(W_{\tilde{s},x}x_t + W_{\tilde{s},h}h_{t-1} + b_{\tilde{s}}) \tag{34}$$

$$i_t = \text{sigmoid}(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \tag{35}$$

Next, the new cell states are determined according to the results from the previous two steps as written below:

$$s_t = f_t \otimes s_{t-1} + i_t \otimes \tilde{s}_t \tag{36}$$

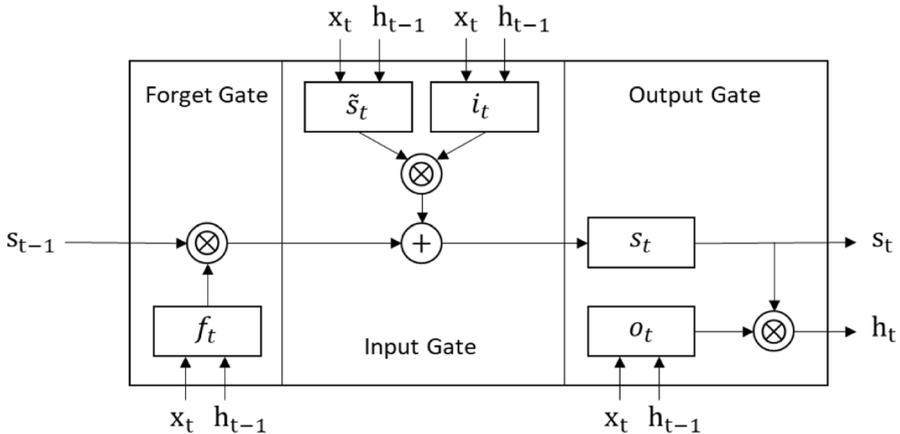


Fig. 12 Schematic of the LSTM model for financial data prediction presented by Fischer and Krauss

where  $\otimes$  is the element-by-element multiplication. Then, the output gate specifies which information from the cell state is used as output as written below:

$$o_t = \text{sigmoid}(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o) \quad (37)$$

$$h_t = o_t \otimes \tanh(s_t) \quad (38)$$

In Eqs. (33) to (38),  $W_{f,x}$ ,  $W_{f,h}$ ,  $W_{\tilde{s},x}$ ,  $W_{\tilde{s},h}$ ,  $W_{i,x}$ ,  $W_{i,h}$ ,  $W_{o,x}$ , and  $W_{o,h}$  are the weight matrices, and  $b_f$ ,  $b_{\tilde{s}}$ ,  $b_i$ , and  $b_o$  are the bias vectors.

Fischer and Krauss in [34] use S&P 500 index from 1989 to 2015 divided to training and trading datasets for training the LSTM and predicting unseen data. They build portfolios of various sizes, trade according to their proposed model, and evaluate the performance of their LSTM model based on performance metrics including daily return, Sharpe ratio, and accuracy. The current study differs from the study by Fischer and Krauss since we do not generate trading rules based on our model, and thus the performance measures such as daily return and Sharpe ratio are not suitable metrics for comparisons in this paper. However, the hybrid model presented in the current study can be compared to the model presented by Fischer and Krauss according to the resulting accuracy of the models. They show that their proposed LSTM model results in an accuracy of 54.3%, whereas the hybrid model by integrating the fundamental and technical analysis presented in this study yields an average accuracy of 70.36%. Moreover, it should be noted that the reason we use data Technology companies is that the hybrid model presented in this study performs based on integrating the fundamental and technical analysis of stock market. Combining data from different sectors (i.e., Technology, Financials, Oil & Gas, etc.) adds more noises to our data since the companies from different sectors may include different financial ratios reported in their financial statements. Therefore, we use data from one sector only in our hybrid model to reduce the noises, whereas the LSTM model presented by Fischer and Krauss performs based on technical analysis only (past historical prices) and does not take into account the fundamental analysis of the stocks. Hence, they use data from S&P 500 regardless of the sector the company belongs to, without increasing the noises. However, for the comparison purposes, we aim to use the same data with the same settings as in [34], and we obtain an average directional accuracy of approximately 59.92%, which is still greater than the accuracy resulting from the proposed LSTM model presented in [34] (i.e., 54.3%).

### 7.3 Hierarchical Graph Attention Network

Prior to presenting the methodology by Kim et al. in [35], no existing work had incorporated the impact of utilizing various types of relations on stock movement prediction or determining an effective way of selectively aggregating information on various relation types. Kim et al. present a hierarchical attention network for stock movement prediction (HATS) to address this issue observed in the literature. Their methodology includes a feature extraction module which performs based on LSTM, as described in the previous subsection, presented in [34], a relational modeling module which

performs based on a graph neural network (GNN), and a prediction layer which predicts the future stock movements.

In the hierarchical attention network, at the first state attention layer, HATS extracts significant information from a set of neighboring nodes. Then, different weights based on the current state of a neighborhood node are calculated. The relation type embedding vector  $e_{r_m}$  and the node representations are examined. The attention score is calculated as:

$$u_{ij} = x_{ij}^{rm} W_s + b_s \tag{39}$$

$$a_{ij}^{rm} = \frac{\exp(u_{ij})}{\sum_k \exp(u_{ik})}, k \in N_i^{rm} \tag{40}$$

where  $x_{ij}^{rm} \in R^{(2f+d)}$  is the concentrated vector;  $W_s \in R^{(2f+d)}$  and  $b_s \in R$  are the parameters. A representation of a vector of relation  $m$  for company  $i$  can be written as:

$$s_i^{rm} = \sum_{j \in N_i^{rm}} a_{ij}^{rm} e_j \tag{41}$$

where  $e_j$  represents the  $j$  th relation-type embedding vector.  $s_i^{rm}$ ,  $e_j$ , and  $e_{r_m}$  are concentrated to utilize  $\tilde{x}_i^{rm}$  as input to the relation attention later as:

$$\tilde{u}_i^{rm} = \tilde{x}_i^{rm} W_r + b_r \tag{42}$$

$$\tilde{a}_i^{rm} = \frac{\exp(\tilde{u}_i^{rm})}{\sum_k \exp(\tilde{u}_i^{rk})} \tag{43}$$

where  $W_r$  and  $b_r$  are the parameters. Then, the weighted vectors of each relation type are added to build an aggregated relation representation:

$$e_i^r = \sum_k \tilde{a}_i^{(rk)} s_i^{rk} \tag{44}$$

The representation of a node is finally added as:

$$\bar{e}_i = e_i^r + e_i \tag{45}$$

Once the HATS is developed, it is used for predicting the stock movements of the companies. The performance metrics used by Kim et al. in [35] for evaluating the performance of their proposed model on portfolios of stocks from different sectors as well as the individual companies are return, Sharpe ratio, and accuracy. Since in the current study we do not build a portfolio for evaluating our proposed hybrid model, and also, we do not generate buy/sell trading strategies, the performance measures such as return and Sharpe ratio are not appropriate metrics for the comparison between our proposed hybrid model and HATKS by Kim et al. as in [35]. However, accuracy can be used for comparison between the performance resulting from

the two models. Moreover, the other main difference between our methodology and the one by Kim et al. is the type of analysis we use. In [35], they use only the historical stock prices (after feature extraction) as the input to their predictive model, while our hybrid model presented in the current study uses the financial ratios of the companies in addition to the historical stock prices as the primary input variables. Hence, collecting data from companies which belong to different sectors will add noises to the data. Therefore, we attempt to collect data from companies which belong to the same sector to reduce noises. However, for comparison purposes, as it was mentioned in the previous subsection, we use the same experiment protocol as in [35]. The numerical results reveal that our proposed hybrid model based on integrating the technical and fundamental analysis of stock market, under the same experiment protocol as in [35], yields an accuracy of approximately 48.36% which is still greater than that of HATS model (i.e., approximately 40% on average). This indicates that the combination of technical and fundamental analysis contains more information as compared with individual models which perform solely based on technical or fundamental analysis.

## 8 Conclusions

Predicting the future stock trends is important for investors who seek to produce a return on the capital invested. This is a complicated and difficult task because the stock market is affected by many factors. AI-based techniques including neural networks have been successfully applied to solve the problems of stock market predictability. In this study, multilayer feedforward perceptron neural network (MLP) is adopted to perform a fundamental and technical analysis and predict the short-term stock trends. In addition, a nonlinear autoregressive with exogenous input (NARX) in neural networks is used to integrate the two primary stock market analyses into a hybrid model. Twelve significant financial ratios are selected by conducting a feature selection preprocess, and the selected ratios are transformed into discretized counterparts by repeating the MLP hyper-parameter optimizations. The number of stock prices as our input data is determined through an autocorrelation preprocess. Moreover, the historical stock prices are categorized into different clusters, and the dimensionality of the input space is reduced using a self-organizing map. A numerical comparison of the hybrid model with regression, ANN, and SVM models are performed, and it is shown that our hybrid model outperforms the other three models in terms of the correlation coefficients on the training and testing datasets. This study shows that fundamental and technical models are capable of predicting the short-term stock trends with certain accuracies of 64.38% and 62.85%, respectively. Stock price prediction can be enhanced by using a hybrid model integrating fundamental and technical analysis resulting in an average directional accuracy of 70.36%, which is greater than those of the state-of-the-art models. These values are very satisfying when compared to the market data. It is also shown that a cross-validation verifies the quality of our hybrid model and the data fitting of this model is precise. The numerical results yield support for the overall efficiency of the hybrid model. It is found that the hybrid model results in the best performance except with the stocks

that have downward trend. The advantages of the hybrid model are as follows: (1) it is easy to implement and operate, and (2) it is not too expensive. In the future work, a supervised clustering method can be studied and compared to the existing (unsupervised) classification systems. The prediction accuracy of our model can also be examined in a portfolio scheme, which has been built based on the selected stocks from the supervised clusters. The proposed model can also be applied to time series in different disciplines and branches of science including physics, biology, etc.

**Acknowledgements** We are grateful to the Reviewers, whose comments helped us improve the paper.

**Funding** Open access funding provided by the University of Strathclyde, Glasgow Scotland UK.

**Data Availability** The datasets analyzed during the current study are available in the Yahoo Finance repository (historical stock prices: <https://finance.yahoo.com/>) and the Nasdaq repository (financial statements: <https://www.nasdaq.com/>). These data were derived from the following public domain resources: <https://finance.yahoo.com/quote/%5EIXIC/>; <https://www.nasdaq.com/market-activity/stocks/ndaq/financials>.

## Declarations

**Conflict of Interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Basu S (1977) Investment performance of common stocks in relation to their price-earnings ratios: a test of the efficient market hypothesis. *J Finance* 32(3):663–682
2. Fama EF (1970) Efficient capital markets: A review of theory and empirical work. *J Finance* 25(2):383–417
3. Blume L, Easley D, O'hara M (1994) Market statistics and technical analysis: the role of volume. *J Finance* 49(1):153–181
4. Brock W, Lakonishok J, LeBaron B (1992) Simple technical trading rules and the stochastic properties of stock returns. *J Finance* 47(5):1731–1764
5. Lo AW, Mamaysky H, Wang J (2000) Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. *J Finance* 55(4):1705–1765
6. Namdari A, Durrani TS (2018) Predictive power of GARCH model against VMA and FMA Trading rules: the case of DJIA and DAX. In Proceedings of the International Annual Conference of the American Society for Engineering Management. (pp. 1–10). American Society for Engineering Management (ASEM)
7. Lev B, Thiagarajan SR (1993) Fundamental information analysis. *J Account Res* 190–215
8. McQueen G, Rokey VV (1993) Stock prices, news, and business conditions. *Rev Financ Stud* 6(3):683–707

9. Bettman JL, Sault SJ, Schultz EL (2009) Fundamental and technical analysis: substitutes or complements? *Accounting & Finance* 49(1):21–36
10. Chan LK, Hamao Y, Lakonishok J (1991) Fundamentals and stock returns in Japan. *J Finance* 46(5):1739–1764
11. Yildiz B, Yezegel A (2010) Fundamental analysis with artificial neural network. *Int J Bus Fin Res* 4(1):149–158
12. Devine K (1995) An examination of quarterly financial ratio stability: implications for financial decision making. *J Appl Bus Res (JABR)* 11(1):81–97
13. Namdari A, Mohammadi S, Drzymalski J, Salmon C (2017) Proposing a self-optimizing neural network to predict total stock return. In *IIE Annual Conference. Proceedings* (pp. 2153–2158). Institute of Industrial and Systems Engineers (IISE)
14. Allen F, Karjalainen R (1999) Using genetic algorithms to find technical trading rules1. *J Finance Econ* 51(2):245–271
15. Tan TZ, Quek C, Ng GS (2005) Brain-inspired genetic complementary learning for stock market prediction. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on* (Vol. 3, pp. 2653–2660). IEEE
16. Hughes J, Liu J, Su W (2008) On the relation between predictable market returns and predictable analyst forecast errors. *Rev Acc Stud* 13(2–3):266–291
17. Namdari A, Li ZS (2018) Integrating fundamental and technical analysis of stock market through multi-layer perceptron. In *2018 IEEE Technology and Engineering Management Conference (TEM-SCON)* (pp. 1–6). IEEE
18. Ou JA, Penman SH (1989) Financial statement analysis and the prediction of stock returns. *J Account Econ* 11(4):295–329
19. Patel J, Shah S, Thakkar P, Kotecha K (2015) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Syst Appl* 42(1):259–268
20. Mantalos P, Karagrigoriou A, Střelec L, Jordanova P, Hermann P, Kiseřák J, ... Stehlík M (2019) On improved volatility modelling by fitting skewness in ARCH models. *J Appl Stat* 1–33
21. Sheta AF, Ahmed SEM, Faris H (2015) A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. *Soft Computing* 7(8):2
22. Kimoto T, Asakawa K, Yoda M, Takeoka M (1990) Stock market prediction system with modular neural networks. In *Neural Networks. IJCNN International Joint Conference on* (pp. 1–6). IEEE
23. Fernandez-Rodriguez F, Gonzalez-Martel C, Sosvilla-Rivero S (2000) On the profitability of technical trading rules based on artificial neural networks: evidence from the Madrid stock market. *Econ Lett* 69(1):89–94
24. Leigh W, Purvis R, Ragusa JM (2002) Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. *Decis Support Syst* 32(4):361–377
25. Roman J, Jameel A (1996) Backpropagation and recurrent neural networks in financial analysis of multiple stock market returns. In *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on* (Vol. 2, pp. 454–460). IEEE
26. Chang PC, Wu JL, Lin JJ (2016) A Takagi-Sugeno fuzzy model combined with a support vector regression for stock trading forecasting. *Appl Soft Comput* 38:831–842
27. Atsalakis GS, Valavanis KP (2009) Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert Syst Appl* 36(7):10696–10707
28. Chen Y, Wang X (2015) A hybrid stock trading system using genetic network programming and mean conditional value-at-risk. *Eur J Oper Res* 240(3):861–871
29. Briza AC, Naval PC Jr (2011) Stock trading system based on the multi-objective particle swarm optimization of technical indicators on end-of-day market data. *Appl Soft Comput* 11(1):1191–1201
30. Wang F, Philip LH, Cheung DW (2014) Combining technical trading rules using particle swarm optimization. *Expert Syst Appl* 41(6):3016–3026
31. Ang KK, Quek C (2006) Stock trading using RSPOP: a novel rough set-based neuro-fuzzy approach. *IEEE Trans Neural Networks* 17(5):1301–1315
32. Chavarnakul T, Enke D (2008) Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Syst Appl* 34(2):1004–1017

33. Boyacioglu MA, Avci D (2010) An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange. *Expert Syst Appl* 37(12):7908–7912
34. Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res* 270(2):654–669
35. Kim R, So CH, Jeong M, Lee S, Kim J, Kang J (2019) Hats: a hierarchical graph attention network for stock movement prediction. arXiv preprint arXiv:1908.07999
36. Dunne RA (2007) A statistical approach to neural networks for pattern recognition (Vol. 702). John Wiley & Sons
37. Romero E, Sopena JM (2008) Performing feature selection with multilayer perceptrons. *IEEE Trans Neural Networks* 19(3):431–441
38. Box GE, Jenkins GM, Reinsel GC, Ljung GM (2015) Time series analysis: forecasting and control. John Wiley & Sons
39. Nair BB, Kumar PS, Sakthivel NR, Vipin U (2017) Clustering stock price time series data to generate stock trading recommendations: an empirical study. *Expert Syst Appl* 70:20–36
40. Nanda SR, Mahanty B, Tiwari MK (2010) Clustering Indian stock market data for portfolio management. *Expert Syst Appl* 37(12):8793–8798
41. Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons
42. Wijaya YB, Kom S, Napitupulu TA (2010) Stock price prediction: comparison of Arima and artificial neural network methods-an Indonesia stock's case. In 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies (pp. 176–179). IEEE
43. Kara Y, Boyacioglu MA, Baykan ÖK (2011) Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. *Expert Syst Appl* 38(5):5311–5319

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.