Expert Systems With Applications 213 (2023) 119203

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Predictability of stock returns using neural networks: Elusive in the long term

Adam Chudziak

SGH Warsaw School of Economics, al. Niepodległości 162, 02-554 Warsaw, Poland

ARTICLE INFO

Keywords:
Stock returns
Neural networks
Financial markets
Forecasting
Time series

ABSTRACT

The rapidly growing neural network literature continually reports successful stock price forecasting results. Many of these studies use relatively short evaluation periods, spanning only a couple of years. In this paper, sustainability of the neural network forecast quality over the long term is analysed. Feedforward and recurrent networks are used to predict the direction of monthly stock price movements, with past price data as predictors. The analysis is conducted on the NYSE stocks over the 1971–2015 period and all the evaluations are performed out-of-sample. Statistically significant directional predictability for selected assets is found. However, the trading simulations reveal that directional predictability does not guarantee trading performance better than the benchmark buy-and-hold strategy. The opportunities for investors to use the tested models for profit appear to be episodic and periodically enhanced e.g. in periods of recession.

1. Introduction

The interest in the problem of stock returns forecasting extends far beyond academic research into professional and private practice. An increased volume of machine learning methods, including artificial neural networks, addresses this well-established and, at the same time, unresolved problem (Jiang, 2021). Noticeably, most of the reported results tend to be overwhelmingly positive, which raises the question of whether the quality of results is sustainable in the long term. This paper undertakes the issue of the long-term viability of neural network forecasting models.

A significant part of the machine learning research uses past price data as predictor variables. It has been shown the seminal paper by Lo et al. (2000), that such predictability can exist, and it is possible to extract some information from the past movements of prices. While this view is in contradiction with the claims of market efficiency, past prices had been widely used in the industry practice. However, it has been pointed out, that the predictability occurring in the financial time series is not a constant, but rather an episodic phenomenon. Pesaran and Timmermann (2002) raised the issue of the potential instability of the price generating process and showed how identifying structural brakes in the financial series can lead to improved market timing. Kolev and Karapandza (2017) demonstrated that the evidence on the predictability of the equity premium depends on the data split, and choosing different splits might lead to contradicting conclusions. Further evidence of instability of return predictions can be found, among others, in Paye and Timmermann (2006), Gonzalo and Pitarakis (2012), or Lettau and Van Nieuwerburgh (2008).

The varying predictability of stock prices has been attributed to the changes in the underlying price generating process (Timmermann, 2008). Since the investors' behaviour influences the prices, the best an individual predictive model can do is to find evidence of temporary predictability. When a forecasting pattern becomes well-established it loses its predictive power. Finding the 'pockets' of predictability has been discussed, among others by Demetrescu et al. (2020).

Meanwhile, the developments in the artificial neural network (ANN) methods for financial markets are accelerating. At least since investigating the IBM daily stock returns by White (1988), there is an interest in forecasting using the artificial neural networks. The field is sprawling in various directions, as the methods are mixed and matched in several ways, see, e.g., Gu et al. (2018), Hafezi et al. (2015), Nabipour et al. (2020), Sang and Di Pierro (2019), Ticknor (2013), Yu et al. (2020). The reason is that composing a neural predictive system is unrestrictive and abundant in choice of networks types, training methods, meta parameters, etc.

Based on the classical understanding of market efficiency, predicting future price movements using past prices as predictor variables should be impossible. Yet, many try to outsmart the market, and method advancements coax investors to try new forecasting techniques. The palette of possibilities is wide, and the changing characteristics of the market, its socioeconomic surroundings, and market regulations disturb the investment practices. This paper is an attempt to take a step back and analyse the long-term usability of neural networks in the market.

Stock price forecasting is performed using different time frames, ranging from intraday trading to yearly forecasts. We focus on the longer time frames, trying to predict prices monthly. The ability to forecast is not only interesting in terms of the development of tools and algorithms, but also from the market efficiency perspective. In

E-mail address: achudz@sgh.waw.pl.

https://doi.org/10.1016/j.eswa.2022.119203
Received 19 January 2022; Received in revised form 7 October 2022; Accepted 31 October 2022
Available online 9 November 2022

the short term, the inefficiencies have been investigated and described by, e.g., (Schulmeister, 2009). The longer-term poses a different kind of challenge for forecasters, as it is generally assumed that in the longer term the markets tend to efficiency. From the asset management perspective, there are different approaches between large financial institutions and individual investors, who tend to trade less frequently, including monthly reviews of investments (Bodie, 2015).

It has been reported that forecasting the sign of the stock return can be more meaningful for the investment practice, then predicting the price itself (Leitch & Tanner, 1991). Furthermore, sign predictability has been reported more frequently, and can exist even without predictability in mean of returns (for discussion see e.g. Christoffersen & Diebold, 2006; Nyberg, 2011). Thus, we focus on evaluating the directional predictions of NYSE stock returns. Several popular neural network topologies are tested, including the Multilayer Perceptron (MLP) and the Long–Short Term Memory (LSTM) networks. The idea of changing market characteristics, expressed, i.a., by Lo (2004), Timmermann (2008), was taken into account and a part of the networks was retrained every year on the most recent data. All the evaluations were performed out-of-sample. It is motivated by the practical aim of this research – evaluating the forecasting rather than explanatory power of the networks – and the fact that neural networks performance in-sample is often non indicative on the quality of the out-of-sample results.

The results suggest sporadic predictability occurring in the series. For some of the stocks, the tested networks have statistically significant predictive power. However, this is not the case for all the networks. The trading simulation suggests that over the entire sample there is little to gain over the buy-and-hold benchmark for the majority of stocks, and the problem of choice of the right forecasting model appears, due to changes in performance in time. In subperiods, however, the trading strategies using neural networks can achieve success. As a consequence, a question about the generality of the neural network results presented in the literature arises.

2. Materials and methods

2.1. Data

As the basis for analysis, daily NYSE stock closing prices are used. The New York Stock Exchange is an established market with a high volume of transactions, which encourages efficiency. Since we want a long period neural network performance analysis, only stocks continuously traded between 1965 and 2015 are considered. The analysis is performed over the 1971–2015 period, the data from the years prior to 1971 is used for initial training. The particular stocks are referred to by their tickers. Data was obtained from the stooq database.[1] Stocks available in the database which had full data for the 1965–2015 period were used in experiments. This allowed avoiding data imputation, and the strictly technical availability criterion removed possible choice bias. Nine stocks used are listed in Table 1. The data was analysed also in subperiods, for expansions and contractions. The economic cycle data came from the National Bureau of Economic Research.[2]

Based on the daily adjusted closing prices, periodic logarithmic returns, weekly and monthly, were computed. The analysis concerns logarithmic returns for two reasons. Firstly, achieving good neural network performance usually requires performing some data scaling. Log-returns are useful in this regard. Secondly, their use is widespread in financial practice and modelling, and we try to emulate investor behaviour. The stock market operates on working days. Thus, by weekly logarithmic returns, the logarithmic returns calculated between the last trading days of consecutive weeks are meant. Usually, these days are Fridays; exceptions occur because of holidays and special events. As the date for the end-of-month logarithmic return, the last day of the month for which the weekly log-return was computed is taken.

[1] https://stooq.pl
[2] https://www.nber.org/cycles.html

Table 1
NYSE stocks traded continuously between 1.1.1965 and 31.12.2015 used in experiments.

Ticker	Company name	Industry
ARNC	Alcoa[a](Arconic Inc.)	Metals
BA	Boeing Co.	Aerospace/Defence
CAT	Caterpillar Inc.	Heavy equipment/Engines
DD	DuPont De Nemours Inc.	Chemicals
DIS	The Walt Disney Co.	Entertainment
GE	General Electric Co.	Multi-industry
HPQ	HP Inc.	Computer hardware
IBM	International Business Machines Corp.	Hardware/Software/Services
KO	The Coca Cola Co.	Drink industry

[a]In the investigated period, this company was traded as Alcoa (AA). In 2016 the company was split into two, Arconic Inc. and Alcoa Corp., with old Alcoa shares (AA) becoming Arconic shares (ARNC) and new Alcoa shares (AA) traded separately.

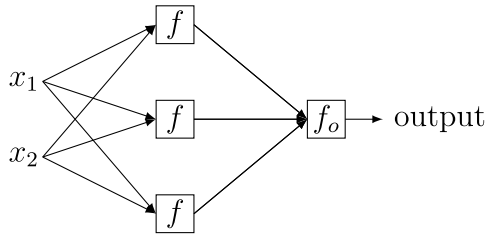

Fig. 1. A simple feedforward neural network. A Multilayer Perceptron with two nodes in the input layer, three nodes in the hidden layer, and a single node in the output layer. The activation function in the output layer f_o can be a sigmoid function, just as in the hidden layer, or any other function, e.g. linear.

2.2. Neural networks

As indicated, the choice regarding the network structure is abundant. The canon of standard methods is vast, and there is room for customisation. In general, an Artificial Neural Network is a computational system built from standardised building blocks. The basic computing unit in the network is called a neuron. The output of one neuron can be given as an input to another. A set of connected neurons forms a neural network; the neurons are organised in layers. The network is called feedforward if the connections between its nodes do not constitute a cycle.

The type of feedforward network used is the *Multilayer Perceptron* (MLP, see Fig. 1). It is characterised by having at least three layers, that is an input layer, an output layer and at least one hidden layer. The connections between neurons are only possible between consecutive layers and in the direction from the input layer to the output layer. A neuron in this network is a $\mathbb{R}^n \to \mathbb{R}$ function, which takes a vector of inputs $x = (x_1, x_2, \ldots, x_n)^t$ and outputs a result of applying a usually nonlinear *activation function* f to the linear combination of inputs and a bias w_0

$$z(x) = f(a(x)) = f\left(\sum_{i=1}^{n} w_i x_i + w_0\right). \tag{1}$$

The networks in this study use the hyperbolic tangent as their activation functions.

Recurrent neural networks use enhanced neurons designed to make use of the temporal structure present in data, such as financial time series or text data. The recurrent neural networks take sequential data as input. To take advantage of that, the RNNs have additional connections between past and present iteration of neurons. A recurrent neuron at the time t in addition to the external input x^t (just as a regular MLP neuron would) takes also a value of a *hidden state* at time $t-1$, which is denoted h^{t-1}. It computes the output z^t and hidden state value to pass into its future iterations. There are many types of recurrent units; they differ in the way they compute and pass their hidden state to their

instance in the next timestep. Two types of recurrent neurons are used — the simple RNN and the Long–Short Term Memory units (Hochreiter & Schmidhuber, 1997).

For the simple RNN, $z^t = h^t$. Then, a simple recurrent neuron additionally to its external input at time t, x^t, takes as an input h^{t-1}, a vector of outputs of the neuron's layer, from the previous timestep

$$z^t = f(a^t(x^t, h^{t-1})) = f\left(w_0 + Wx^t + Uh^{t-1}\right), \tag{2}$$

where W and U are vectors of parameters (weights).

The LSTM neurons (Hochreiter & Schmidhuber, 1997) were constructed to combat problems which occurred in the training of simple RNNs. They add internal loops which prevent gradient decay in training. In a LSTM unit, a *cell state* c_i^t is passed to the unit's next iteration. So, the input of an LSTM unit consists of the external input x^t, the hidden input from the previous instances of neurons in the layer h^{t-1}, and the cell state c_i^{t-1}. After all the processing is done in a cell, it returns its output $z_i^t = h_i^t$ and stores its cell state c_i^t.

The passage of data through time in a LSTM cell is controlled by sigmoid gates. There are three logistic (σ) gates and one hyperbolic tangent gate (tanh). The new addition to the standard RNN model is the cell state passed from the previous instance of the LSTM cell. Its impact on processing at time t is controlled by the *forget gate* $\mathbf{f}_i^t$ defined

$$\mathbf{f}_i^t = \sigma\left(w_{i,0}^{\mathbf{f}} + \sum_{j=1}^{n} w_{i,j}^{\mathbf{f}} x_j^t + \sum_{k=1}^{I_l} u_{i,k}^{\mathbf{f}} h_k^{t-1}\right). \tag{3}$$

At the same time, the impact of input is controlled by the *input gate*

$$\mathbf{i}_i^t = \sigma\left(w_{i,0}^{\mathbf{i}} + \sum_{j=1}^{n} w_{i,j}^{\mathbf{i}} x_j^t + \sum_{k=1}^{I_l} u_{i,k}^{\mathbf{i}} h_k^{t-1}\right), \tag{4}$$

while the input is squashed by the hyperbolic tangent function of the *external input gate*

$$\mathbf{g}_i^t = \tanh\left(w_{i,0}^{\mathbf{g}} + \sum_{j=1}^{n} w_{i,j}^{\mathbf{g}} x_j^t + \sum_{k=1}^{I_l} u_{i,k}^{\mathbf{g}} h_k^{t-1}\right). \tag{5}$$

The cell state is updated based on the previous state and new information

$$c_i^t = \mathbf{f}_i^t c_i^{t-1} + \mathbf{i}_i^t \mathbf{g}_i^t. \tag{6}$$

Basing on the new cell state, the output h_i^t is computed. It is a product of the cell state squashed with hyperbolic tangent and the *output gate* $\mathbf{o}_i^t$ controlling the magnitude of output

$$\mathbf{o}_i^t = \sigma\left(w_{i,0}^{\mathbf{o}} + \sum_{j=1}^{n} w_{i,j}^{\mathbf{o}} x_j^t + \sum_{k=1}^{I_l} u_{i,k}^{\mathbf{o}} h_k^{t-1}\right), \tag{7}$$

$$h_i^t = \mathbf{o}_i^t \tanh\left(c_i^t\right). \tag{8}$$

The internal parameters of the networks, the *weights*, are determined by minimising the prediction error function; the mean squared error was used as the error function for training the networks. The minimisation is usually done using one of the gradient descent optimisation techniques. There is a choice in selecting the gradient descent scheme. In this study the *rmsprop* scheme is used, one of the more popular choices. To prevent overfitting the training set, early stopping was implemented.

2.3. Forecasting models

Two forecasting approaches were used in this study. In the first approach, the networks were trained and retained the same parameters over the entire evaluation sample. In the second approach, the networks were retrained periodically, to keep the parameters up to date with the current market characteristics (Lo, 2004; Timmermann, 2008). The retraining was done yearly, using the last five years of data, in accord with the paradigm of the changing market.

Determining the right neural network structure is, in general, a difficult task. In practice, the choice is often made using grid search validation. Even relatively simply structured MLPs, with only one hidden layer, are universal approximators (Cybenko, 1989; Hornik et al., 1989). However, a satisfactory approximation of the desired function may require a large number of neurons. Also, despite this theoretical result, the way to adjust weights for an optimal approximation is unclear. The empirical research on the optimal architecture for time-dependent problems is ongoing. There have been attempts to assess what is necessary and which of the numerous variants of recurrent networks are optimal, see, e.g. Greff et al. (2015), Jozefowicz et al. (2015). Alas, no architecture proved to be universally best across different applications.

As noted by Gu et al. (2018), it is "unrealistic and unnecessary to find the optimal network by searching over uncountably many architectures". Following their example, a number of architectures sufficient to illustrate the point was considered. Three types of network were used: the feedforward network (MLP), the simple RNN (RNN), and the LSTM network (LSTM). These types of networks were chosen as they have often been employed for monthly stock price forecasts, and remain a common choice today, see for example Cipiloglu Yildiz and Yildiz (2022), Teng et al. (2020), Zhang et al. (2018). For each of the networks types, there are several hidden layer structures. The activation functions were hyperbolic tangents.

Each of the models returns as the output a single value — the forecast of the next month logarithmic return. The sign of this predicted return is the directional forecast. As the input, the eight previous weekly logarithmic returns are taken. The time covered by the input data amounts almost to the last two months of data. For example, at the end of February, the network predicts monthly logarithmic return for March using the eight most recent weekly logarithmic returns counting from the end of February. A month is enough to form a price pattern (Lo et al., 2000). Thus, by taking eight weeks, the patterns occurring in the past month and a half can be caught. We follow the assumption, common among machine learning researchers in finance, that the price patterns might contain information allowing to predict future trends of stock price movements. The evaluation period is 1971–2015, as the first five years of the sample are used solely for training.

2.4. Model evaluation

The output of the forecasting model is $\hat{y}$, the predicted monthly log-return of a stock. We use the directional accuracy (DA, Eq. (9)) to quantify the quality of our predictions. Due to the binary characteristic of our forecasts this hit-or-miss approach is natural. The no predictability accuracy is expected at 50% level.

$$\mathrm{DA} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\mathrm{sgn}(\hat{y}_i)=\mathrm{sgn}(y_i)} \tag{9}$$

A crucial metric for any investor is the return one gets on investment. A simple trading simulation is performed to assess the quality of forecasts using the return on investment metric. We imagine an investor who at the end of each month performs an analysis and decides whether he should own the stock for the next month. To simplify the matter, the investor trades on a single stock instead of constructing a portfolio. The investor's decision is based solely on the forecast. If the forecast sign is positive, the investor buys or continues to own the stock, if not, he sells or continues to stay out of the market. No alternative investment is provided. If the market timing forecasts are correct, then the strategy should allow outperforming the buy-and-hold benchmark.

The market timing test proposed by Pesaran and Timmermann (1992) is used to evaluate the statistical predictability. Its null hypothesis states that the difference between the correct prediction ratio obtained by the tested model is not different from the ratio which would

be achieved if forecasts and the true value of the positive/negative indicator were independent. The PT-test statistic is

$$PT = \frac{\hat{P} - \hat{P}_*}{\sqrt{\widehat{var}(\hat{P}) - \widehat{var}(\hat{P}_*)}}, \tag{10}$$

where $\hat{P}$ is the hit rate, or proportion of agreeing signs in the forecast; $\hat{P}_*$ is the estimated hit rate proportion under the null hypothesis. Asymptotically, the distribution of the PT statistic is normal.

The forecasts generated by different networks are additionally compared using the Diebold-Mariano test (DM, Diebold & Mariano, 1995). It is a nonparametric, model-independent test. Thus, it can be used for comparing forecasts coming from any source: an econometric model, a neural network or a financial analyst. The null hypothesis of the Diebold-Mariano test states

$$\mathrm{H}_0 : \mathrm{E}\left[g\left(e_{1t}\right)\right] = \mathrm{E}\left[g\left(e_{2t}\right)\right], \tag{11}$$

where $\{e_{1t}\}_{t=1,2,\dots T}$ and $\{e_{2t}\}_{t=1,2,\dots T}$ are the errors of two forecasts, and $g : \mathbb{R} \to \mathbb{R}$ is a function of e_{it}.[3] Denoting the sample period size with T, and the loss differential $d_t = g(e_{1t}) - g(e_{2t})$, then if d_t is covariance stationary and short memory and $\hat{f}_d(0)$ is a consistent estimator of the spectral density of d_t at frequency 0, the statistic for this test,

$$DM = \frac{\bar{d}\sqrt{T}}{\sqrt{2\pi \hat{f}_d(0)}}, \tag{12}$$

has an asymptotically standard normal distribution.

3. Results

3.1. Networks trained once

An ideal scenario for a trader using neural networks would be to train a network once and then use it for a long time. For each of the log return series separate models are trained, 66 network topologies for each stock, 594 models in total. Tables A.7–A.9 present the Directional Accuracy for the MLP, RNN, and LSTM networks respectively. The results over the long run are placed around 50% for all the models. This indicates a 'no free lunch' scenario, where a satisfactory forecastability might not be obtainable using the ANNs.

The fact that one method has higher directional accuracy than the other can be an occurrence of a random chance event. It can be for example a result of the discrete character of the DA. Using a non-parametric forecasting ability test, such as the Pesaran–Timmermann test, is a way of verifying whether a model can actually forecast. Tables A.10–A.12 contain the p-values for the PT test for all the used networks. For the majority of the networks the test fails to reject the null hypothesis of independence of forecast and the actual log return. It is worth noting, however, that that is not the case for all the models. In addition, for some stocks, like CAT, for example, there seems to be more predictability than for the rest.

From the practitioner point of view, the ultimate test for the forecasts is whether one can make a profit using them as the basis for trading decisions. We consider only the models for which the PT test rejected the null of no forecasting power to filter out the random profits made by chance. Table 2 presents the ROI for the models where the null for the PT test was rejected with 95% confidence, and the return on investment was higher than benchmark buy-and-hold ROI for the entire sample. Imposing this condition limits the number of successful predictions over the long period to ten cases.

In the analysis up till now all the models were evaluated over the entire evaluation sample. Now, the period 1971–2015 was split in half.

[3] We define the loss function to be the function of the forecast errors. In principle, the loss can be defined to be any function $g(y_t, \hat{y}_t)$ of the true values y_t and forecast $\hat{y}_t$, in which case $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$.

Table 2
Return-on-investment (ROI) for the networks for which the ROI was higher than benchmark buy-and-hold ROI and the PT test rejected the null with 95% confidence. The return on investment on the entire 1971–2015 period is presented in the last column (roiTotal). Columns roi1 and roi2 present the return on investment in the first and second half of the 1971–2015 period respectively.

Network_type	Ticker	Layers	roi1	roi2	roiTotal
LSTM	CAT	3	2.26	0.57	1.29
LSTM	CAT	6	3.92	0.56	2.18
LSTM	CAT	3, 3	3.00	0.61	1.83
LSTM	CAT	3, 2	2.70	0.63	1.70
LSTM	CAT	5, 2	3.19	0.42	1.34
LSTM	CAT	6, 6, 2	0.94	1.22	1.14
LSTM	CAT	6, 4, 2	1.98	0.56	1.10
LSTM	CAT	6, 2, 2	2.22	0.66	1.46
LSTM	HPQ	5	0.57	2.09	1.19
LSTM	HPQ	6	0.87	1.45	1.25
LSTM	IBM	2, 2, 2	2.45	0.51	1.24
RNN	CAT	3	3.03	0.57	1.74
RNN	CAT	6	1.74	0.64	1.12

The first half was treated as the validation set and the second as the testing set, to check if it is possible to pick one of the models from the group which exhibited statistically significant predictive power over the 1971–2015 period. The relative ROI for the halves of the evaluation period are presented in the Table 2. It is interesting, that none of the models which performed better than the benchmark on the validation set could beat the buy-and-hold on the testing set. So, in the end, the trader would be worse off using the model, which performed well on the first half of the sample, than using buy-and-hold. On the other hand, the models which could beat the buy-and-hold on the second part of the sample, would not be chosen based on the performance on the first part of the sample.

3.2. Periodically retrained networks

The temporal changes in the forecast quality showed in the previous section suggest that maybe the models should be updated, so that the newest information is taken into account. In this section we consider networks that are trained every year on the past five years of data. Selected network topologies from Section 3.1. were used; they are denoted by the acronym of the network type followed by numbers indicating the hidden layer structure. That is, for example, MLP62 is a feedforward network with six neurons in the first hidden layer and two neurons in the second hidden layer.

The directional accuracy over the entire sample for each of the stocks, and each of the tested networks, is again around 50% (Table 3). The percentages differ between assets. The highest directional accuracy is exhibited for the KO stock, with the directional accuracy above 55% achieved by all the models. On the other hand, in the worst case, the HPQ stock proved to be particularly difficult to predict, and the directional accuracy of no model exceeded 50%.

The achieved ROI usually is lower than the benchmark buy-and-hold ROI, as shown in Table 4, which contains the portfolio ROI as a fraction of the benchmark buy-and-hold return on investment. For the entire sample, only three models performed better than the benchmark strategy, and all of them achieved this for the same stock. For the other assets, the results are much worse. What is interesting, the highest accuracy percentages did not correspond to the highest relative ROI, even for the strategies outperforming the benchmark.

The forecasting ability measured with the PT-test is statistically significant with 95% confidence for selected models for five out of nine stocks (Table 5). Worth noting is that for the KO stock, all the models showed statistically significant forecasting ability and presented directional accuracy in the 55%–60% (Table 3) interval, and yet their ROI relative to the buy-and-hold was below 0.5 (Table 4). This again signals the difficulties with the choice of the right network for a particular stock.

Table 3
Out-of-sample Directional Accuracy of forecasts over the entire sample of the networks retrained every year. The rows are labelled with the acronym of the network type followed by digits corresponding to the number of neurons in consecutive layers of that network.

	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
MLP6	45.93	48.70	55.19	50.93	49.07	54.63	53.33	53.70	50.56
MLP62	46.85	48.89	55.56	50.93	49.81	54.07	52.96	54.63	52.41
MLP662	47.96	49.26	55.00	51.11	49.26	54.07	53.89	52.78	52.41
RNN6	47.41	47.96	55.93	51.48	48.15	53.89	52.78	52.41	53.15
RNN62	49.07	48.70	54.81	51.30	49.63	52.78	50.19	51.67	49.26
RNN662	46.48	48.33	56.85	49.07	49.26	53.89	52.96	52.22	51.85
LSTM6	52.22	51.48	58.33	51.85	47.59	53.70	52.22	52.41	50.37
LSTM62	51.85	52.04	55.00	52.59	47.59	50.93	55.56	50.93	49.81
LSTM662	51.30	50.93	55.74	51.67	47.96	50.19	55.56	52.96	49.81

Table 4
Out-of-sample trading simulation Return on Investment (ROI), relative to the buy-and-hold ROI for the networks retrained every year. The rows are labelled with the acronym of the network type followed by digits corresponding to the number of neurons in consecutive layers of that network.

	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
MLP6	0.01	0.17	0.23	0.28	0.03	1.00	0.38	0.34	0.27
MLP62	0.01	0.22	0.22	0.21	0.03	1.10	0.31	0.48	0.47
MLP662	0.02	0.33	0.17	0.32	0.03	0.83	0.42	0.17	0.52
RNN6	0.01	0.15	0.28	0.52	0.03	0.61	0.16	0.24	0.54
RNN62	0.03	0.21	0.14	0.30	0.06	1.12	0.06	0.12	0.01
RNN662	0.02	0.15	0.33	0.10	0.05	1.27	0.33	0.24	0.07
LSTM6	0.12	0.43	0.36	0.33	0.07	0.50	0.18	0.39	0.19
LSTM62	0.07	0.43	0.17	0.44	0.04	0.27	0.34	0.27	0.16
LSTM662	0.08	0.21	0.30	0.28	0.05	0.27	0.53	0.29	0.14

Table 5
The p-values of the Pesaran–Timmermann test for the networks retrained every year. The rows are labelled with the acronym of the network type followed by digits corresponding to the number of neurons in consecutive layers of that network.

	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
MLP6	1.00	0.96	0.00	0.57	0.88	0.00	0.04	0.02	0.51
MLP62	1.00	0.96	0.00	0.58	0.74	0.01	0.07	0.00	0.12
MLP662	1.00	0.92	0.01	0.52	0.86	0.01	0.01	0.08	0.11
RNN6	1.00	0.99	0.00	0.44	0.97	0.01	0.14	0.15	0.05
RNN62	1.00	0.97	0.03	0.54	0.80	0.08	0.94	0.44	0.90
RNN662	1.00	0.99	0.00	0.89	0.77	0.01	0.05	0.22	0.21
LSTM6	0.91	0.75	0.00	0.66	1.00	0.02	0.95	0.42	0.69
LSTM62	0.93	0.72	0.10	0.56	1.00	0.62	0.01	0.90	0.89
LSTM662	0.98	0.91	0.00	0.84	1.00	0.73	0.00	0.29	0.93

3.3. Economic cycle impact

The results indicate that training networks on the more recent data does not suffice for getting good forecasts. Perhaps the quality of forecasts can be associated with the economic cycle. The Fig. 2 presents directional accuracies for the models evaluated in Section 3.2, split by the economic expansion or contraction periods. The data is benchmarked against 50% and the percentage of the "upward" movements of stock prices in the periods. In general, the directional accuracy is lower in contractions. However, the results tend to be better relative to the second benchmark. It is particularly apparent for the IBM stock, for which in contractions the majority of models achieved an over 60% directional accuracy. For the comparison of expansion and contraction data, it is worth remembering, that in the second half of the 20th century US economy was in recession sporadically and for short periods. As a consequence, there is less recession data to infer from.

3.4. Pockets of predictability

All that said, with a particularly chosen period and stock, success can be achieved. Table 6 and Fig. 3 present results of trading on the DIS stock from 1996 to 2006 using the MLP6. It is a typical result that could be showcased as a successful implementation of neural network trading. The benchmark return on investment over ten years was 29%, while the MLP6 network achieved 81%. The directional accuracy of 57.5% was enough to beat the benchmark. Furthermore, using the PT test, we can observe that the result is statistically significant at the less than 2% level. Showing just this result, however, obscures the difficulty of the a priori model choice.

Table 6
An example of what could pass as a successful forecasting experiment in a subperiod. Results of the 1-stock (DIS) portfolio trading according to the MLP6 recommendations over the 1996–2006 decade. Statistically significant predictability, indicated using the PT test and the return on investment better than the benchmark by over 50 percentage points. Fig. 3 shows plots of the portfolio values.

	Directional accuracy	ROI	Benchmark ROI	PT statistic	PT p-value
DIS	57.5%	81%	29%	2.12	0.017

To further analyse changing forecasting performance over time we can inspect the five-year moving average directional accuracy of models presented in Fig. 4. The data in the figure suggests that there is no one model which performed best over a long period. However, it stands out that in some periods, for example, around the year 2000, an increased directional accuracy can be observed. The recessions are marked in the figure by the grey background shade. Interestingly, the periods of contraction (in particular 1981–1982 recession) mark the start of an increase of mean directional accuracy for certain stocks, such as KO, DD, HPQ or GE. High directional accuracy over a period means that the market timing of the model over this particular period is heightened, which should be advantageous for trading. Being able to identify such periods before they occur, if possible, would be useful for traders.

4. Discussion

The notorious difficulty of forecasting stock returns makes the development of new methods enticing. On the other hand, in the established, efficient markets, one expects little, if any, predictability. It is particularly difficult if one restricts oneself to the data from within the market. However, even usage of exogenous predictors might not help, as often the out-of-sample performance of the models is poor and unstable (Welch & Goyal, 2008).

In this context, any statistically significant predictability detected can be (and frequently is) presented as a satisfactory result. The results show that for the majority of the tested stocks at least one of the models exhibited statistically significant predictability, and even outperform the buy-and-hold benchmark. However, the difficulty of model choice remains, as the quality of predictions is not stable over time. It is not clear, what is the criterion for the future performance of the neural network model. Retraining the neural network does not resolve this issue. The inequivalence of satisfactory forecasting performance of a model and the yielded return on investment invalidates the possibility of using the proposed networks as the sole predictive model. The goal of any forecasting strategy is to help the investor make rational decisions, and hence the tested models can be used rather as an auxiliary tool.

Although in the long run the profitability is not granted, it was shown that the neural network models can exhibit predictive power, and thus provide additional information for investors. The outputs of a neural network could be viewed as a weak predictor, and, in this context, used as a part of a predictive model. The use of many weak predictors can improve forecasting performance and has been recently investigated in the context of stock price predictions (Zhang et al., 2019).

In this study, only the past price data for forecasting was used. It may be beneficial to use other predictors in a neural forecasting framework. Alternatively, perhaps, a more advanced feature processing could enhance the predictions. The advances in network design also

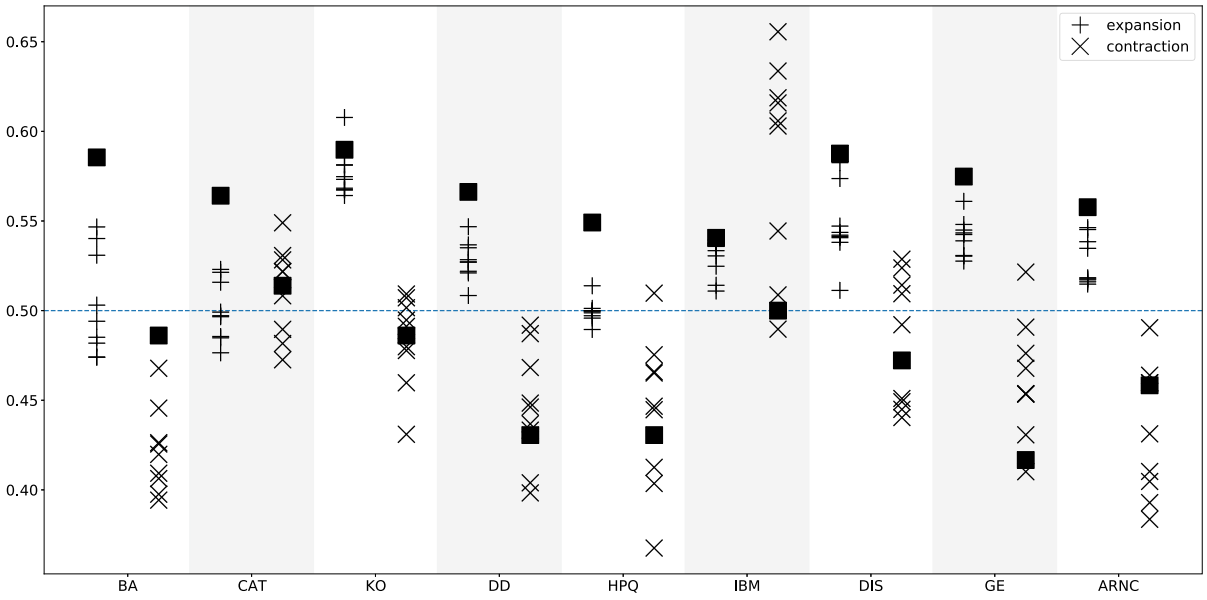

Fig. 2. Directional Accuracy, split by stock, for all the expansions and contractions for the networks retrained every year. Expansions are marked with a "+", contractions are marked with a "x". The full black squares represent the percentage of "up" movements of the stock in the tested periods. A small random noise was added to the directional accuracy values to separate the markers in the figure.

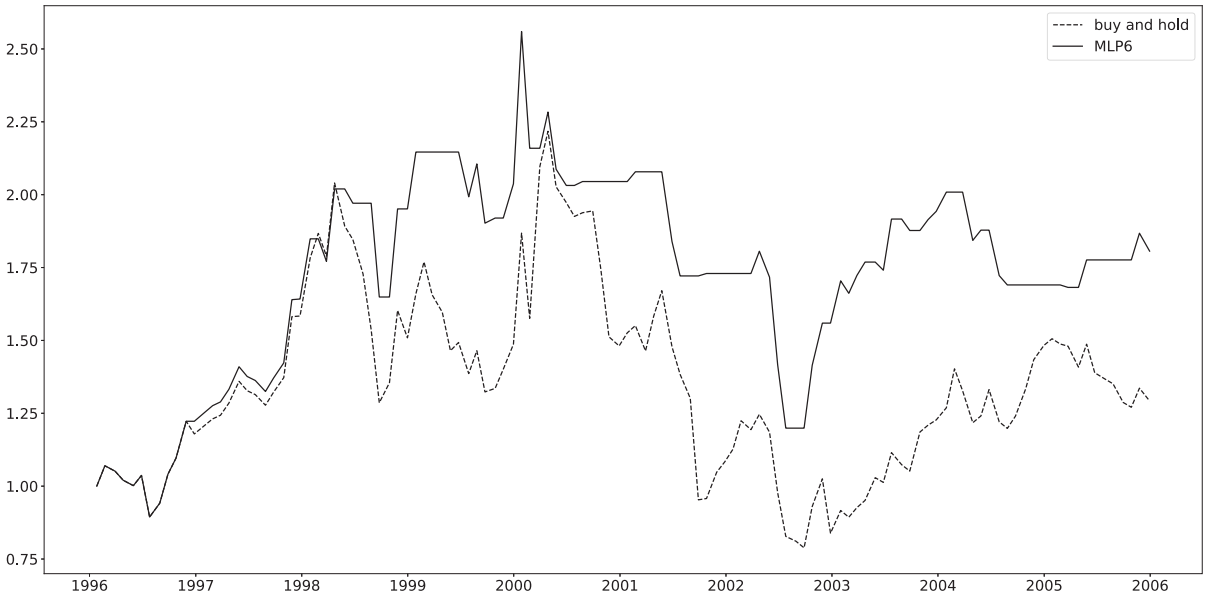

Fig. 3. The $1, 1-stock (DIS) portfolio trading according to the MLP6 recommendations and the benchmark $1 buy-and-hold portfolio for the 1996–2006 period.

offer new tools at forecasters' disposal. However, the number of choices one has to make while constructing a forecasting neural model makes it impractical to check every possible topology and poses a real difficulty for practical application. The model choice, as well as feature design, remain the main difficulty. Each of these issues is explored in detail in the neural network literature. However, the research in the field is sprawling in several directions, widening the choice, rather than formulating definitive conclusions. For practitioners, the development of a systematic approach to the problem of stock forecasting using neural networks would be beneficial.

The opportunities to achieve higher profits might lie elsewhere. We should note that the long-term nature of the study required analysing only stocks quoted long enough. Any study within this framework has a similar implicit survivorship bias. It is possible that price movements of well-established stocks are harder to predict. On a similar note, the New York Stock Exchange is a big, well-established and liquid market with many trading agents; All these qualities suggest market efficiency and thus fewer possibilities of achieving excessive profit. Furthermore, only large-cap NYSE stocks were investigated. It is possible that while in a developed, efficient market there are fewer pockets of predictability, there are more possibilities when trading on stocks of companies with smaller market capitalisation, in a less-developed (thus — less efficient) market. Such stocks tend to exhibit higher volatility, which might be possible to exploit, if only periodically.

The lack of long term profitability illustrates the generalisability problem of the models. It is likely, that the tested networks cannot extract the information at all times, but only in periods. The problem thus lies in the identification of the correct period, asset and

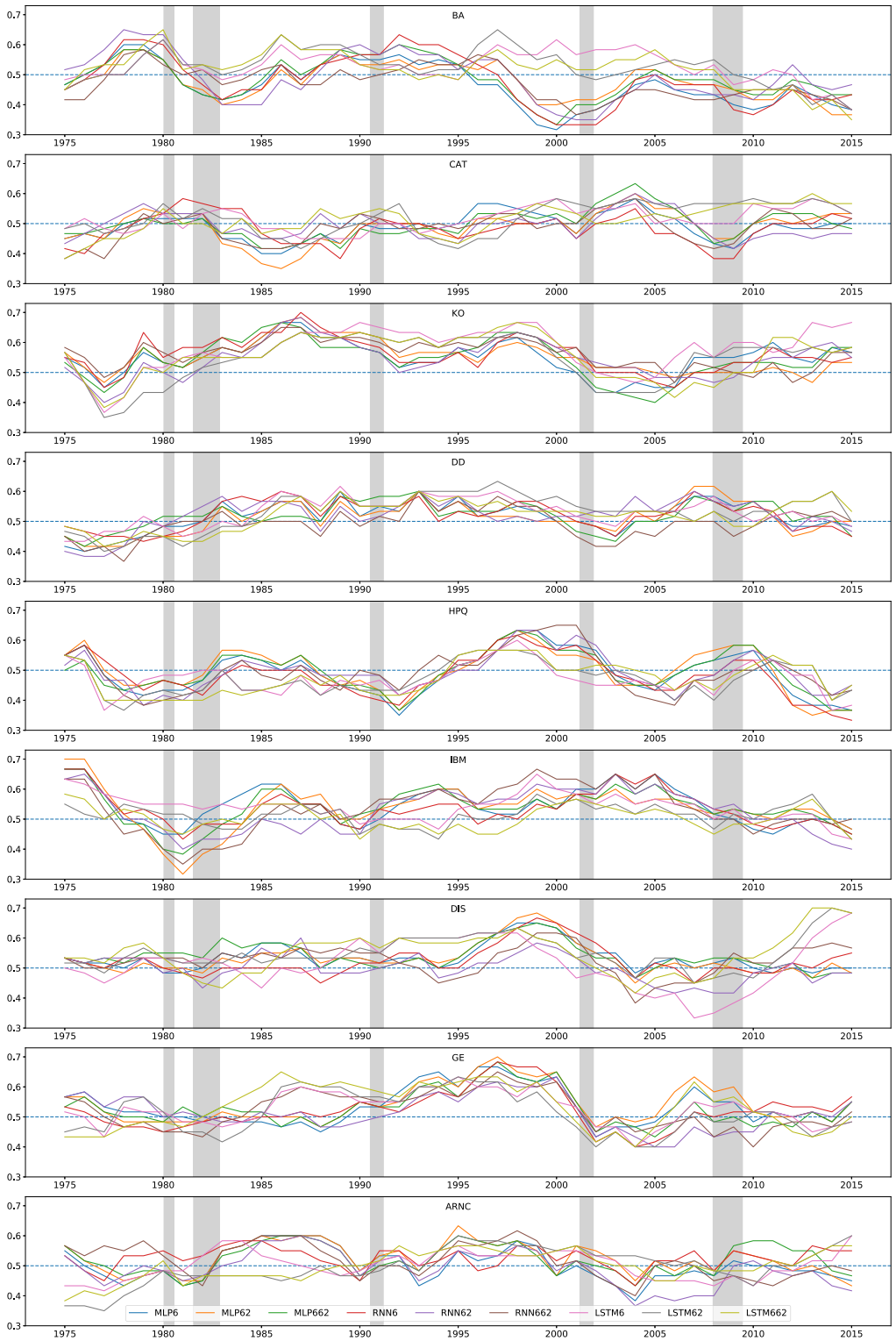

Fig. 4. Five-year mean directional accuracy of models retrained yearly. Each of the plots presents data for one stock over the entire sample period. The horizontal blue line is the benchmark 50%. The periods in grey are periods of economic contraction, according to NBER.

model capable of exploiting the information (Demetrescu et al., 2020; Timmermann, 2008). Analysed in isolation, the results obtained in subperiods could suggest overly positive conclusions. It is possible, that the results are due to the episodic nature of the forecasting power of the models. As a result, the reported successes of neural networks in stock price forecasting might not translate into practical solutions. From the practitioner perspective, consistent quality of the method is desired. Supplementing the forecasting model with a module that identifies the predictability periods, e.g. based on economic factors external to the market, is a potential solution. Devising methods of identification of the predictability periods would be a natural extension of this study.

Table A.7
Out-of-sample Directional Accuracy (DA) of predictions of 198 MLP networks. The first column (Layers) informs on the number of neurons in consecutive layers of the network. The next columns correspond to different stocks. For each stock a separate model is trained. The DA is expressed as a percentage and rounded to two digits.

Layers	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
1	47.78	50.56	53.70	47.96	54.81	45.00	51.67	47.96	50.00
2	48.52	49.44	54.07	46.30	52.59	45.56	53.70	48.52	50.93
3	48.52	51.85	54.44	48.33	53.15	46.11	54.26	48.89	50.37
4	48.89	48.89	53.89	48.33	52.78	44.63	54.63	49.81	49.63
5	48.33	50.93	54.44	47.22	53.15	44.81	52.59	48.33	49.44
6	49.44	49.81	55.37	47.22	52.78	45.37	52.04	48.89	49.63
1, 1	48.33	53.15	54.44	47.78	51.67	47.96	54.81	50.00	48.70
2, 2	48.70	49.81	52.22	47.96	53.15	47.04	55.00	50.56	48.52
3, 3	49.81	49.07	54.26	47.78	53.15	45.00	55.56	51.30	48.89
3, 2	48.70	48.70	55.74	49.26	52.59	44.26	54.81	49.44	49.63
4, 4	50.93	53.52	55.19	49.81	52.78	46.67	52.96	51.48	50.56
4, 2	48.33	51.30	53.70	48.70	53.33	45.19	54.81	51.11	50.93
5, 5	48.52	51.11	54.63	48.15	52.96	45.37	52.96	50.93	49.63
5, 2	48.70	48.89	54.07	47.96	53.33	45.00	55.37	48.33	48.15
6, 6	52.59	49.44	54.44	47.04	53.15	44.81	53.15	50.56	50.19
6, 4	48.52	50.93	55.56	48.52	52.59	46.48	55.37	48.89	49.44
6, 2	48.33	52.22	54.81	47.78	52.78	47.04	55.19	50.93	47.41
6, 6, 2	49.44	48.89	54.07	47.41	53.33	46.85	54.63	48.70	49.81
6, 4, 4	49.63	47.96	57.41	49.26	52.22	45.37	55.19	48.89	49.26
6, 4, 2	49.26	49.26	54.63	47.59	52.96	45.00	52.04	49.07	50.74
6, 2, 2	51.11	51.85	54.26	47.96	52.59	45.00	55.19	50.19	48.33
2, 2, 2	47.96	48.89	53.33	46.85	53.33	44.81	56.30	48.52	50.00

Table A.8
Out-of-sample Directional Accuracy (DA) of predictions of 198 RNN networks. The first column (Layers) informs on the number of neurons in consecutive layers of the network. The next columns correspond to different stocks. For each stock a separate model is trained. The DA is expressed as a percentage and rounded to two digits.

Layers	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
1	47.78	55.00	57.59	53.52	52.04	51.48	55.00	49.63	49.44
2	51.48	54.81	57.78	44.81	51.85	50.19	56.30	49.07	48.15
3	47.78	55.00	53.70	51.48	51.48	50.37	55.93	49.44	46.48
4	48.89	53.70	54.44	48.89	54.07	47.96	54.63	48.89	49.07
5	47.41	53.33	54.63	50.56	53.15	47.78	52.96	48.70	49.44
6	48.33	53.15	51.30	51.30	52.78	48.52	52.59	48.70	49.07
1, 1	52.96	54.81	57.96	50.56	53.33	53.70	57.04	50.19	52.59
2, 2	47.59	55.56	57.22	50.00	53.33	48.70	54.44	49.81	46.30
3, 3	51.11	51.11	55.00	44.63	54.07	47.41	51.30	50.56	50.37
3, 2	50.93	53.70	57.04	47.04	46.30	49.63	51.30	50.56	44.63
4, 4	46.67	52.22	52.59	47.78	48.52	48.33	49.44	50.19	49.63
4, 2	48.33	50.93	58.15	51.11	51.11	48.33	52.96	49.07	50.19
5, 5	48.52	52.41	50.19	51.11	48.89	47.78	52.59	49.44	49.44
5, 2	50.00	53.15	52.96	50.00	53.33	46.48	49.07	49.26	47.96
6, 6	49.07	49.81	52.04	47.59	50.56	48.70	52.04	50.74	50.00
6, 4	50.56	52.22	51.30	47.04	46.48	47.59	47.04	51.67	50.00
6, 2	45.93	53.89	52.22	46.67	50.00	47.41	53.52	48.52	49.07
6, 6, 2	47.22	53.70	52.41	46.67	47.41	51.30	50.93	48.15	47.96
6, 4, 4	51.11	52.22	52.22	48.70	53.15	52.04	47.96	48.70	50.19
6, 4, 2	51.11	53.70	53.33	48.52	52.22	49.26	48.89	49.44	50.37
6, 2, 2	52.04	53.52	51.11	45.19	52.59	47.78	54.26	48.52	48.52
2, 2, 2	47.78	53.33	56.30	52.04	53.70	47.96	54.07	49.26	47.04

Table A.9
Out-of-sample Directional Accuracy (DA) of predictions of 198 LSTM networks. The first column (Layers) informs on the number of neurons in consecutive layers of the network. The next columns correspond to different stocks. For each stock a separate model is trained. The DA is expressed as a percentage and rounded to two digits.

Layers	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
1	53.15	55.74	57.78	47.96	52.04	51.48	54.81	51.67	50.19
2	53.33	54.44	57.78	45.93	51.67	51.67	54.26	49.63	50.74
3	53.15	54.07	56.67	44.63	50.19	52.22	52.78	49.44	50.00
4	50.93	54.63	54.63	46.30	54.26	51.30	52.96	49.81	51.67
5	51.85	53.89	52.59	43.52	53.70	49.81	53.15	48.52	50.74
6	50.37	55.19	54.44	45.19	55.00	51.48	50.37	50.00	51.11
1, 1	51.48	54.81	57.59	50.56	52.78	50.56	57.04	55.37	50.00
2, 2	52.96	53.89	57.59	44.81	52.59	53.70	52.41	49.63	50.74
3, 3	51.48	55.00	57.78	45.00	50.56	51.11	52.41	51.30	50.37
3, 2	47.59	55.74	53.33	44.44	52.59	50.93	51.11	53.15	50.19
4, 4	49.81	53.89	55.74	42.78	52.59	54.63	54.26	50.19	50.00
4, 2	52.41	55.93	57.22	45.74	52.22	55.00	52.59	49.26	50.00
5, 5	52.22	55.74	54.81	45.37	52.96	51.48	51.11	47.96	51.30
5, 2	50.00	55.74	53.33	43.15	51.11	55.19	54.44	50.00	50.19
6, 6	46.11	55.00	54.26	43.89	50.93	50.93	56.11	49.26	51.85
6, 4	46.11	55.56	51.85	42.96	49.26	50.93	56.85	49.26	50.00
6, 2	50.93	53.89	55.19	47.59	51.48	53.70	53.89	50.56	49.07
6, 6, 2	48.70	55.19	56.85	45.56	52.59	54.26	58.15	52.59	50.19
6, 4, 4	52.96	54.26	56.30	47.41	52.78	54.26	48.52	51.11	50.74
6, 4, 2	51.48	54.44	56.11	44.44	51.48	53.70	56.67	55.37	49.07
6, 2, 2	51.11	55.56	53.15	47.22	52.78	52.41	53.33	52.04	50.74
2, 2, 2	47.59	55.19	57.59	45.19	51.67	54.07	57.22	50.93	49.63

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Science Center, Poland, grant no. 2016/23/N/HS4/01897.

Appendix A. Detailed results for models trained once

This appendix contains the Directional Accuracy and the Pesaran–Timmermann test results for the once-trained models, see Tables A.7–A.9.

Appendix B. ARIMA comparison

One of the established approaches to time series analysis and forecasting is to use the Autoregressive Integrated Moving Average models (ARIMA). In general, an ARIMA(p, d, q) model of variable y_t can be written as

$$y_t = C + \sum_{i=1}^{p} \varphi_i y'_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}, \tag{B.1}$$

where y' denotes the d times differenced series y_t, ϵ_t is the error at time t, and C, φ_i, θ_j are the parameters. As an additional check of the ANN forecast quality we compare the results to the forecasts of the ARIMA model.

Every month in the testing period, for each of the tested stock, the ARIMA model was estimated on the monthly stock price data. The estimation sample was, similarly to the retrained networks case, the last five years of data; the difference is that a new ARIMA model was estimated each month, so at the end of February 1975 the estimation sample spanned from march 1970 to February 1975. Then the estimated model was used for a one-month-ahead forecast. The directional accuracy of the forecasts was computed as well as the return on investment of the simulated trading strategy, analogous to the one used with the ANN predictions. The results are presented in Table B.13.

In general, there is no improvement on the neural network predictions. In fact, for each stock there is a neural network that performed better than the ARIMA model, both in terms of directional accuracy and trading performance. The notable exception is the ARNC stock, for which a relatively high (while still worse than the benchmark buy-and-hold) ROI was achieved, with directional accuracy equal to only 52,13%. The ARIMA models were able, unlike most of the ANN, to correctly identify the largest price movements over the evaluation sample. Nonetheless, the trading performance is still worse than the benchmark buy-and-hold strategy.

Table A.10
P-values for the Pesaran–Timmermann test for 198 MLP networks. The first column (Layers) informs on the number of neurons in consecutive layers of the network.

Layers	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
1	1.00	0.81	0.67	1.00	0.00	1.00	1.00	0.57	0.42
2	0.99	0.95	0.18	1.00	0.10	1.00	0.91	0.51	0.25
3	0.92	0.59	0.02	0.94	0.04	0.98	0.81	0.36	0.53
4	0.93	0.97	0.36	1.00	0.07	1.00	0.77	0.33	0.43
5	0.98	0.81	0.06	1.00	0.05	1.00	0.97	0.48	0.46
6	0.98	0.82	0.30	0.99	0.07	0.99	0.99	0.38	0.42
1, 1	0.93	0.08	0.36	1.00	0.17	0.96	0.81	0.29	0.60
2, 2	0.99	0.94	0.39	1.00	0.05	0.96	0.74	0.22	0.56
3, 3	0.99	0.95	0.06	1.00	0.04	1.00	0.63	0.16	0.77
3, 2	0.99	0.98	0.00	1.00	0.09	1.00	0.72	0.35	0.56
4, 4	0.95	0.15	0.31	0.99	0.09	0.98	0.97	0.13	0.51
4, 2	0.92	0.73	0.03	0.93	0.04	1.00	0.78	0.17	0.26
5, 5	0.87	0.79	0.04	0.97	0.07	1.00	0.96	0.29	0.52
5, 2	0.95	0.97	0.45	1.00	0.04	1.00	0.70	0.48	0.61
6, 6	0.51	0.94	0.09	0.99	0.04	1.00	0.94	0.29	0.44
6, 4	0.99	0.79	0.31	1.00	0.12	0.98	0.72	0.43	0.58
6, 2	0.92	0.37	0.33	1.00	0.07	0.83	0.72	0.24	0.67
6, 6, 2	0.98	0.98	0.58	0.99	0.04	0.93	0.73	0.36	0.63
6, 4, 4	0.98	0.91	0.07	1.00	0.14	1.00	0.78	0.43	0.50
6, 4, 2	0.99	0.98	0.04	0.94	0.05	1.00	0.99	0.42	0.45
6, 2, 2	0.87	0.62	0.30	1.00	0.08	1.00	0.65	0.33	0.58
2, 2, 2	0.94	0.81	0.06	0.98	0.03	1.00	0.60	0.46	0.49

Table A.11
P-values for the Pesaran–Timmermann test for 198 RNN networks. The first column (Layers) informs on the number of neurons in consecutive layers of the network.

Layers	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
1	0.35	0.59	0.50	0.57	0.21	0.81	0.52	0.48	0.57
2	0.35	0.63	0.45	0.96	0.14	0.85	0.42	0.56	0.86
3	0.99	0.03	0.83	0.92	0.29	0.75	0.45	0.50	0.88
4	0.95	0.22	0.04	0.58	0.01	0.99	0.82	0.61	0.73
5	0.91	0.14	0.31	0.95	0.08	0.92	0.84	0.72	0.66
6	0.99	0.04	0.52	0.86	0.07	0.88	0.96	0.93	0.39
1, 1	0.16	0.66	0.39	0.45	0.04	0.40	0.31	0.46	0.20
2, 2	0.38	0.07	0.17	0.94	0.25	0.66	0.85	0.68	0.78
3, 3	0.94	0.95	0.63	0.94	0.01	0.98	0.98	0.28	0.43
3, 2	0.68	0.10	0.00	0.99	0.66	0.72	1.00	0.19	0.64
4, 4	0.83	0.36	0.07	0.95	0.81	0.93	1.00	0.48	0.47
4, 2	0.92	0.56	0.25	0.82	0.65	0.96	0.95	0.53	0.37
5, 5	0.55	0.33	0.51	0.95	0.67	0.99	0.36	0.79	0.91
5, 2	0.60	0.28	0.16	0.97	0.11	0.99	1.00	0.75	0.90
6, 6	0.78	0.33	0.39	0.98	0.54	0.89	0.36	0.22	0.44
6, 4	0.95	0.23	0.59	0.99	0.94	0.94	1.00	0.18	0.39
6, 2	0.90	0.04	0.92	1.00	0.48	0.93	0.84	0.85	0.59
6, 6, 2	0.72	0.05	0.09	0.98	0.72	0.71	0.99	0.93	0.52
6, 4, 4	0.76	0.56	0.26	0.99	0.05	0.27	1.00	0.45	0.63
6, 4, 2	0.40	0.13	0.05	1.00	0.22	0.97	1.00	0.92	0.71
6, 2, 2	0.18	0.22	0.65	0.99	0.08	0.98	0.57	0.82	0.66
2, 2, 2	0.65	0.59	0.03	0.20	0.36	0.79	0.89	0.65	0.75

Appendix C. Alternative trading rule

This appendix presents the results of the trading simulation using an alternative strategy, that is, short selling. In the main body of the text, the trading strategy quit the market for the time when the models forecasted a price decrease. Table C.14 is analogous to Table 2. It contains the return on investment for the trading strategy relative to the benchmark buy-and-hold for the models for which the PT test rejected the hypothesis of lack of predictive ability, and the relative ROI is more than one. These are the same models as in Table 2. Additionally, there is the ROI for each half of the sample included. The conclusions are similar to before, the magnitude of return of investment is increased by short selling. Still, the performance in the first half of the sample is not indicative of the performance of the second part of the sample. Table C.15 presents the results for the retrained networks, so it is analogous to Table 4. There is no apparent qualitative difference between the results.

Table A.12
P-values for the Pesaran–Timmermann test for 198 LSTM networks. The first column (Layers) informs on the number of neurons in consecutive layers of the network.

Layers	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
1	0.14	0.50	0.45	0.95	0.21	0.79	0.70	0.72	0.55
2	0.04	0.07	0.20	1.00	0.24	0.72	0.15	0.57	0.58
3	0.13	0.01	0.01	0.94	0.81	0.48	0.26	0.69	0.34
4	0.74	0.29	0.01	1.00	0.01	0.63	0.31	0.59	0.32
5	0.60	0.08	0.55	1.00	0.03	0.63	0.36	0.98	0.21
6	0.65	0.00	0.09	1.00	0.00	0.68	0.85	0.47	0.29
1, 1	0.32	0.14	0.50	0.91	0.42	0.86	0.46	0.50	0.37
2, 2	0.25	0.06	0.50	1.00	0.39	0.13	0.39	0.52	0.67
3, 3	0.35	0.02	0.45	1.00	0.78	0.60	0.29	0.29	0.39
3, 2	0.84	0.01	0.33	1.00	0.49	0.62	0.57	0.11	0.31
4, 4	0.84	0.31	0.06	1.00	0.49	0.22	0.15	0.28	0.74
4, 2	0.44	0.10	0.34	0.99	0.59	0.14	0.41	0.72	0.52
5, 5	0.53	0.14	0.25	1.00	0.16	0.63	0.51	0.69	0.62
5, 2	0.90	0.01	0.41	1.00	0.62	0.06	0.14	0.33	0.94
6, 6	0.85	0.14	0.34	1.00	0.77	0.27	0.19	0.71	0.11
6, 4	0.91	0.01	0.83	1.00	0.98	0.81	0.05	0.83	0.42
6, 2	0.50	0.39	0.04	0.95	0.74	0.09	0.10	0.44	0.60
6, 6, 2	0.79	0.01	0.50	1.00	0.49	0.13	0.22	0.04	0.41
6, 4, 4	0.39	0.25	0.43	1.00	0.45	0.21	0.78	0.17	0.26
6, 4, 2	0.30	0.03	0.28	1.00	0.78	0.40	0.33	0.50	0.95
6, 2, 2	0.58	0.01	0.43	0.96	0.34	0.59	0.27	0.36	0.35
2, 2, 2	1.00	0.20	0.50	1.00	0.73	0.04	0.50	0.42	0.92

Table B.13
Quality of forecasts obtained from rolling ARIMA models.

	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
Directional accuracy	48.98	49.17	52.13	45.66	49.54	50.28	50.65	51.57	52.13
Relative ROI	0.07	0.09	0.08	0.05	0.02	0.37	0.25	0.11	0.90

Table C.14
Return-on-investment (ROI) for the networks for which the ROI was higher than benchmark buy-and-hold ROI and the PT test rejected the null with 95% confidence. The return on investment on the entire 1971–2015 period is presented in the last column (roiTotal). Columns roi1 and roi2 present the return on investment in the first and second half of the 1971–2015 period respectively. The trading strategy allowed short selling, in contrast with results presented in Table 2.

Network_type	Ticker	Layers	roi1	roi2	roiTotal
LSTM	CAT	3	5.12	0.32	1.66
LSTM	CAT	6	15.37	0.31	4.77
LSTM	CAT	3, 3	8.97	0.38	3.37
LSTM	CAT	3, 2	7.30	0.39	2.88
LSTM	CAT	5, 2	10.17	0.18	1.79
LSTM	CAT	6, 6, 2	0.88	1.48	1.30
LSTM	CAT	6, 4, 2	3.90	0.31	1.21
LSTM	CAT	6, 2, 2	4.93	0.43	2.13
LSTM	HPQ	5	0.33	4.36	1.43
LSTM	HPQ	6	0.75	2.09	1.57
LSTM	IBM	2, 2, 2	5.98	0.26	1.54
RNN	CAT	3	9.18	0.33	3.02
RNN	CAT	6	3.04	0.41	1.25

Table C.15
Out-of-sample trading simulation Return on Investment (ROI), relative to the buy-and-hold ROI for the networks retrained every year. The rows are labelled with the acronym of the network type followed by digits corresponding to the number of neurons in consecutive layers of that network. The trading strategy allowed short selling, in contrast with results presented in Table 4.

BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
−0.00	0.01	0.05	0.07	−0.01	0.99	0.14	0.11	0.03
−0.00	0.04	0.05	0.03	−0.01	1.22	0.09	0.22	0.20
−0.00	0.10	0.02	0.10	−0.01	0.69	0.18	0.03	0.25
−0.00	0.01	0.07	0.26	−0.01	0.36	0.02	0.05	0.28
−0.00	0.03	0.01	0.08	−0.00	1.24	−0.00	0.01	−0.07
−0.00	0.01	0.10	−0.01	−0.01	1.62	0.11	0.05	−0.06
0.01	0.17	0.13	0.10	−0.00	0.24	0.03	0.15	−0.01
0.00	0.18	0.02	0.19	−0.01	0.05	0.11	0.07	−0.02
0.00	0.03	0.09	0.07	−0.01	0.05	0.28	0.08	−0.03

Table D.16
Diebold–Mariano test statistics testing differences between forecasts generated using retrained models. Cases for which the difference is statistically significant at 10% significance level are presented in bold font.

	BA	CAT	KO	DD	HPQ	IBM	DIS	GE	ARNC
MLP6 vs MLP62	−0.83	0.11	−0.41	−0.03	**1.87**	−0.91	−0.22	1.11	1.26
MLP6 vs MLP662	−0.18	1.17	0.06	−0.91	0.48	−0.46	−1.13	0.43	0.53
MLP6 vs RNN6	1.43	**2.43**	**1.80**	**2.83**	**3.14**	0.43	0.25	0.93	1.46
MLP6 vs RNN62	**1.97**	0.23	1.07	**2.47**	**2.46**	0.52	0.96	0.68	−0.25
MLP6 vs RNN662	**−1.83**	−0.13	−0.02	−1.26	0.53	**−1.74**	−1.40	−0.89	1.27
MLP6 vs LSTM6	**4.71**	**4.14**	**4.61**	**4.25**	**3.62**	**1.93**	**3.17**	**2.60**	**3.83**
MLP6 vs LSTM62	**4.58**	**3.96**	**4.30**	**4.37**	**3.76**	1.45	**3.27**	**2.50**	**3.99**
MLP6 vs LSTM662	**4.72**	**4.04**	**4.18**	**4.05**	**4.00**	1.32	**3.56**	**2.34**	**3.66**
MLP62 vs MLP662	0.60	1.23	0.34	−0.96	−1.15	0.48	−1.36	−0.47	−0.70
MLP62 vs RNN6	**1.67**	**2.52**	**1.92**	**3.08**	**2.19**	0.90	0.34	0.54	0.89
MLP62 vs RNN62	**2.36**	0.20	1.20	**2.53**	**2.12**	0.87	0.95	0.37	−0.73
MLP62 vs RNN662	−1.60	−0.17	0.13	−1.23	0.13	−1.46	−1.33	−1.25	0.61
MLP62 vs LSTM6	**5.06**	**4.06**	**4.64**	**4.39**	**3.49**	**2.30**	**3.20**	**2.28**	**3.44**
MLP62 vs LSTM62	**4.91**	**3.95**	**4.41**	**4.50**	**3.60**	**1.79**	**3.32**	**2.20**	**3.65**
MLP62 vs LSTM662	**4.95**	**4.04**	**4.33**	**4.14**	**3.81**	**1.66**	**3.61**	**2.04**	**3.35**
MLP662 vs RNN6	1.43	**1.95**	1.46	**3.97**	**2.57**	0.59	0.93	0.83	1.11
MLP662 vs RNN62	**2.00**	−0.23	0.88	**2.98**	**2.23**	0.61	1.30	0.62	−0.49
MLP662 vs RNN662	**−1.81**	−0.53	−0.05	−1.05	0.40	**−1.72**	−1.00	−1.03	0.76
MLP662 vs LSTM6	**4.44**	**3.77**	**4.73**	**4.73**	**3.70**	**1.98**	**3.35**	**2.49**	**3.14**
MLP662 vs LSTM62	**4.35**	**3.66**	**4.38**	**4.79**	**3.82**	1.52	**3.48**	**2.37**	**3.39**
MLP662 vs LSTM662	**4.51**	**3.79**	**4.31**	**4.42**	**4.09**	1.39	**3.79**	**2.19**	**3.13**
RNN6 vs RNN62	0.72	**−1.94**	−0.46	0.87	−0.40	0.24	0.69	−0.14	−1.24
RNN6 vs RNN662	**−2.42**	**−2.30**	−1.22	**−2.28**	−0.82	**−1.82**	**−1.69**	−1.39	−0.14
RNN6 vs LSTM6	**4.19**	**2.81**	**3.92**	**3.38**	**2.93**	**1.89**	**2.95**	**1.91**	**2.47**
RNN6 vs LSTM62	**4.34**	**2.60**	**3.73**	**3.61**	**3.13**	1.39	**3.10**	**1.81**	**2.71**
RNN6 vs LSTM662	**4.53**	**2.71**	**3.51**	**3.29**	**3.34**	1.21	**3.44**	**1.67**	**2.43**
RNN62 vs RNN662	**−3.03**	−0.26	−0.81	**−2.90**	−0.65	−1.60	**−2.02**	−1.36	1.08
RNN62 vs LSTM6	**3.08**	**4.24**	**3.94**	**3.36**	**2.90**	**2.04**	**3.10**	**1.72**	**2.70**
RNN62 vs LSTM62	**3.24**	**4.00**	**3.69**	**3.88**	**3.11**	1.24	**3.33**	1.63	**2.86**
RNN62 vs LSTM662	**3.29**	**4.11**	**3.58**	**3.42**	**3.28**	1.12	**3.82**	1.50	**2.61**
RNN662 vs LSTM6	**5.34**	**4.47**	**3.78**	**4.77**	**2.28**	**2.51**	**3.45**	**2.27**	**2.82**
RNN662 vs LSTM62	**5.15**	**4.38**	**3.69**	**5.14**	**2.35**	**2.24**	**3.69**	**2.34**	**2.94**
RNN662 vs LSTM662	**5.40**	**4.69**	**3.50**	**5.06**	**2.50**	**2.07**	**4.10**	**2.22**	**2.56**
LSTM6 vs LSTM62	0.81	−0.00	−0.88	**1.77**	0.54	−1.03	0.47	0.22	1.02
LSTM6 vs LSTM662	1.31	0.08	**−1.69**	1.05	0.69	−1.16	1.28	−0.38	−0.00
LSTM62 vs LSTM662	0.87	0.13	−1.37	−0.50	0.38	−0.39	1.32	−0.98	−1.50

Appendix D. Differences between forecasts

This appendix contains the results of the Diebold-Mariano test for forecasts of the models retrained every year. The test statistics are tabulated in Table D.16, statistically significant differences (at 10% level) are marked in bold in the table.

One can observe a pattern in the results. The significant differences occur most of all between different types of networks. Within a particular type it is more common that the predictions are not statistically different, albeit not in all cases, e.g. forecasts of MLP6 and MLP62 are statistically different for HPQ. This pattern occurs for all of the tested networks. The results reinforce the idea that it is worth trying several forecasting approaches, as the yielded results differ.

References

Bodie, Z. (2015). Thoughts on the future: Life-cycle investing in theory and practice. *Financial Analysts Journal*, *71*, 43–48.

Christoffersen, P. F., & Diebold, F. X. (2006). Financial asset returns, direction-of-change forecasting, and volatility dynamics. *Management Science*, *52*(8), 1273–1287.

Cipiloglu Yildiz, Z., & Yildiz, S. B. (2022). A portfolio construction framework using LSTM-based stock markets forecasting. *International Journal of Finance & Economics*, *27*(2), 2356–2366, Publisher: John Wiley & Sons, Ltd.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314.

Demetrescu, M., Georgiev, I., Rodrigues, P. M. M., & Taylor, A. M. R. (2020). Testing for episodic predictability in stock returns. *Journal of Econometrics*.

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

Gonzalo, J., & Pitarakis, J. Y. (2012). Regime-specific predictability in predictive regressions. *Journal of Business & Economic Statistics*, *30*(2), 229–241.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2015). LSTM: A search space Odyssey. arXiv:1503.04069 [cs].

Gu, S., Kelly, B., & Xiu, D. (2018). *Empirical asset pricing via machine learning*: *Working paper 25398*, National Bureau of Economic Research.

Hafezi, R., Shahrabi, J., & Hadavandi, E. (2015). A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price. *Applied Soft Computing*, *29*, 196–210.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, *184*, Article 115537.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd international conference on machine learning*.

Kolev, G. I., & Karapandza, R. (2017). Out-of-sample equity premium predictability and sample split–invariant inference. *Journal of Banking & Finance*, *84*, 188–201.

Leitch, G., & Tanner, J. E. (1991). Economic forecast evaluation: Profits versus the conventional error measures. *The American Economic Review*, *81*(3), 580–590.

Lettau, M., & Van Nieuwerburgh, S. (2008). Reconciling the return predictability Evidencethe review of financial studies: Reconciling the return predictability evidence. *The Review of Financial Studies*, *21*(4), 1607–1652.

Lo, A. W. (2004). The adaptive markets hypothesis. *The Journal of Portfolio Management*, *30*(5), 15–29.

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The Journal of Finance*, *55*(4), 1705–1765.

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., & S., S. (2020). Deep learning for stock market prediction. *Entropy*, *22*(8), 840, Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

Nyberg, H. (2011). Forecasting the direction of the US stock market with dynamic binary probit models. *International Journal of Forecasting*, *27*(2), 561–578.

Paye, B. S., & Timmermann, A. (2006). Instability of return prediction models. *Journal of Empirical Finance*, *13*(3), 274–315.

Pesaran, M. H., & Timmermann, A. (1992). A simple nonparametric test of predictive performance. *Journal of Business & Economic Statistics*, *10*(4), 461–465.

Pesaran, M., & Timmermann, A. (2002). Market timing and return prediction under model instability. *Journal of Empirical Finance*, *9*(5), 495–510.

Sang, C., & Di Pierro, M. (2019). Improving trading technical analysis with tensorflow long short-term memory (LSTM) neural network. *The Journal of Finance and Data Science*, *5*(1), 1–11.

Schulmeister, S. (2009). Profitability of technical stock trading: Has it moved from daily to intraday data? *Review of Financial Economics*, *18*(4), 190–201.

Teng, H. W., Li, Y. H., & Chang, S. W. (2020). Machine learning in empirical asset pricing models. In *2020 International conference on pervasive artificial intelligence* (pp. 123–129). Taipei, Taiwan: IEEE.

Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, *40*(14), 5501–5506.

Timmermann, A. (2008). Elusive return predictability. *International Journal of Forecasting*, *24*(1), 1–18.

Welch, I., & Goyal, A. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, *21*(4), 1455–1508.

White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns. In *IEEE international conference on neural networks*: *vol. 2*, (pp. 451–458). San Diego, CA, USA: IEEE.

Yu, Z., Qin, L., Chen, Y., & Parmar, M. D. (2020). Stock price forecasting based on LLE-BP neural network model. *Physica A*, *553*, Article 124197.

Zhang, R., Huang, C., Zhang, W., & Chen, S. (2018). Multi factor stock selection model based on LSTM. *International Journal of Economics and Finance*, *10*(8), 36.

Zhang, Y., Zeng, Q., Ma, F., & Shi, B. (2019). Forecasting stock returns: Do less powerful predictors help? *Economic Modelling*, *78*, 32–39.