



Event prediction within directional change framework using a CNN-LSTM model

Ahoora Rostamian¹ · John G. O'Hara¹

Received: 21 June 2021 / Accepted: 28 July 2022
© The Author(s) 2022

Abstract

Financial forecasting has always been an intriguing research area in the field of finance. The widely accepted approach to forecast financial data is to perform predictions using time series data. In time series analysis, sampling the financial data with a predefined frequency (e.g. hourly, daily) leads to an uneven and discontinued data flow. Directional Change is a newly proposed approach that replaces physical time within the financial data and establishes an event-driven framework. With the emergence of the machine and deep learning-based methods, researchers have utilised them in financial time series. These techniques have shown to outperform conventional approaches. This paper aims to employ the CNN-LSTM model to investigate its predictive competence within the Directional Change (DC) framework to predict DC event prices. To obtain this objective, we first create the tick bars/candles of the GBPUSD, EURUSD, USDCHF, and USDCAD tick prices from January to August 2019. Then, the DC-based summaries of the selected tick bar/candle for each currency pair will be generated and fed to the CNN-LSTM model. The CNN-LSTM network architecture incorporates the robustness of Convolutional Neural Network (CNN) in feature extraction and Long Short-Term Memory (LSTM) in predicting sequential data. The results suggest that the performance of the CNN-LSTM model improves significantly within the DC framework.

Keywords Directional change framework · Event prediction · Price prediction · CNN · LSTM

1 Introduction

Although predicting a financial asset price has been an intriguing area of research, it has proven to be a highly complex task due to the inherent complexity, volatility, and nonlinearity of financial markets. The widely accepted approach to analyse financial data is time-series analysis. Conventionally, in order to analyse the financial time series, prices are recorded by sampling data points at fixed time intervals (Daily, weekly, monthly). Researchers first decide how often to sample the data in this method, and then they take snapshots at the chosen frequency.

Consequently, financial time series are unevenly spaced and discontinuous concerning the flow of physical time [8]. Thereby, the interval-based summary of the price may miss important key events and lose profitable trade opportunities.

To tackle the aforementioned shortcoming of the traditional approach of time series analysis, Guillaume et al. [10] proposed a new method for scaling time. Directional changes (DC) is an alternative approach that replaces the notion of “physical time scale” and looks beyond the physical time constraints within financial data, and constitutes an event-driven approach. Hence, market data are being observed from the event-based rather than the interval-based perspective. With the recent success of machine and deep learning approaches, many researchers have applied various algorithms and architectures on financial time series to predict financial assets' price and movement [27]. Mehtab and Sen [19] presented a suite of CNN-based regression models with a high level of accuracy and robustness in forecasting multivariate financial

✉ Ahoora Rostamian
ar19738@essex.ac.uk

John G. O'Hara
johara@essex.ac.uk

¹ Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, Colchester, Essex, United Kingdom

time series. This study proposes a deep learning-based regression model to predict the price of the directional change framework events the currency pairs in the foreign exchange (FX) market and evaluate its performance within and without the Directional Change framework.

The remainder of this research paper is organised as follows. Section 2 presents a brief overview of the related work in the field of financial forecasting. Section 3 presents the methodology of this study, which includes discussing the directional change framework, Long Short-Term Memory, Convolutional Neural Networks, Support Vector and Random Forest regression, data, experiment and results. Finally, in Sect. 4, we conclude the paper.

2 Related work

Financial forecasting has always been an exciting research area in the financial industry. Numerous studies have been published on machine learning models with relatively better performances than classical time series forecasting techniques [17, 29, 30, 34, 37]. Researchers endeavoured to use nonlinear models to predict. With the advent of machine learning methods such as neural networks, support vector machines (SVM), researchers utilise them for time series prediction [16]. Zbikowski [38] employed Volume-Weighted SVM feature selection techniques to enhance classifier accuracy to create a stock trading strategy. Choudhury et al. [4] utilised k-means and SVR to predict market volatility and prices for two days in the Indian stock market. Artificial neural networks (ANNs), a sub-class of machine learning models are widely used for predictive data-mining tasks. The applicability of artificial neural networks to stock market predictions was first hypothesised by White [36], with some indications of success by Saad et al. [25]. Artificial neural networks, in essence, mimic the structure of biological neural networks where neurons are interconnected and learn from experience.

In 2003, Zhang used neural network and auto-regressive integrated moving average model (ARIMA) to forecast stocks. The experimental results proved the advantage of neural networks in nonlinear data forecasting [39]. Abu Hammad et al. [1] investigated the Jordanian stock market with a multi-layer back propagation (BP) network, nonetheless did not discuss the BP proneness to fall into a local minimum. Zhang et al. [40] proposed a stock forecasting model based on LM-BP neural network which improves the traditional BP neural network. Wang et al. [35] proposed a wavelet neural network to forecast stock prices. Persio and Honchar [6] compared the performance of three different variants of RNNs to predict Google's stock price. Their model showed better results for LSTM

compared to the basic RNN and the Gated Recurrent Unit (GRU), with an accuracy of 72% within a five day period. They shuffled the train and test data to prevent the network from over-fitting.

The prediction of the Nifty Index movements using the open, high, low, close prices was implemented with an LSTM RNN architecture in Roondiwala et al. [24] work. Their work reached a root mean squared error of 0.0086 after training with 500 epochs. Karmiani et al. [13] compared the performance of LSTM to SVM, backpropagation and Kalman filter with epochs between 10 to 100 and found that LSTM has high accuracy and low variance. Fischer and Krauss [5] performed a large-scale prediction of S and P500 from December 1992 to October 2015 and showed that the LSTM model outperforms the machine learning methods and deep networks. Nelson et al. [20] proposed an LSTM-based model in combination with 175 technical indicators to predict the stock market movement. Salis et al. [26] presented a thorough investigation of the application of LSTM models and artificial neural networks in predicting the fluctuation of daily gold prices. Zhuge et al. [41] predicted the opening stock prices using their proposed LSTM model. They combined the classification results and the analysis of the naive Bayesian-based emotions. In 2018, Hu [12] used CNN to predict time series. Their results showed that CNN can predict time series, however, the forecasting accuracy is relatively low. Sezer and Ozbayoglu [28] utilised the CNN model to classify the daily price of Dow 30 stocks and Exchange-Traded Funds (ETFs).

3 Methodology

The methodology is structured as follows. In Sect. 3.1, the directional change framework will be introduced. Sections 3.2 and 3.3 explain Long-Short Term Memory (LSTM) and Convolutional Neural networks (CNNs). Section 3.4 briefly introduces Support Vector and Random Forest regression. Sections 3.5 and 3.6 describe the data and the Average True Range. Finally, in Sect. 3.7, the experiment will be presented in detail.

3.1 Directional change framework

The directional Change (DC) is an approach to summarise price movement by transforming a time series price curve into an intrinsic time curve [32]. Under the DC framework, a DC event is identified by a substantial change in the price of an asset, defined as a price change greater than a pre-defined threshold value θ . Following a DC event, an overshoot (OS) event happens until the next DC event in the opposite direction. Figure 1 illustrates a time series and

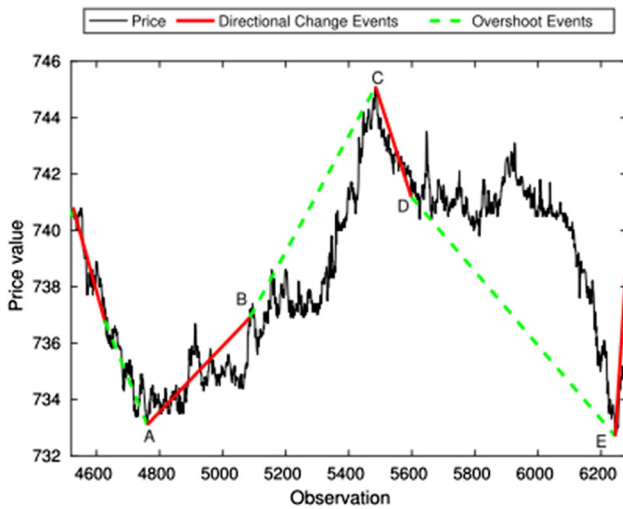


Fig. 1 A share price and the corresponding intrinsic time curve for $\theta = 0.05\%$, for a selected time period

the corresponding intrinsic time series for a $\theta = 0.01\%$. Based on DC approach, the market is broken down into an alternating uptrend and downtrend. An upturn event indicates that the price change between the current market price p_t and the last low price p_l is greater than a threshold θ :

$$p_t \geq p_l(1 + \theta) \tag{1}$$

As illustrated in Fig. 1, the move from point A to B is an upturn DC event. By the same token, a downturn event is defined as an event where the difference between the current price p_t and the last high price p_h is lower than a fixed threshold θ [32]:

$$p_t \leq p_h(1 - \theta) \tag{2}$$

A trend ends whenever a price change of the same threshold θ is observed in the opposite direction, see [2]. It should be noted that different thresholds generate different series of events. The notion of using different thresholds is that each threshold might be considered significant by a different trader. Smaller thresholds create more directional changes compared to larger ones. As it was mentioned above the value of the threshold needs to be predetermined when summarising price movements using the DC. It represents how big of a price change the observer considers as significant.

Tsang and Chen [31], Bakhach et al. [2], and Golub et al. [9] have explored classical machine learning techniques such as the Hidden Markov Model and Naïve Bayes classifier to predict the behaviour of tick prices within an event-driven approach in the directional change framework. In our work, we extended their work into a deep neural network paradigm. Since different thresholds generate different market summaries, we also proposed

incorporating the Average True Range indicator to determine the DC thresholds dynamically. For the interested reader, a more detailed discussion on Directional Change may be found in [3].

Algorithm 1: Defining directional-change (DC) and overshoot (OS) events.

Result: DC-based event summaries

Initialization:(event=upturn

$p_h = p_l = p_0, \theta \geq 0, t_0^{dc} = t_1^{dc} = t_0^{os} = t_1^{os} = t_0$);

if event is upturn **then**

if $p_t \leq p_h * (1 - \theta)$ **then**

 event \leftarrow Downturn Event

$p_l \leftarrow p_t$

$t_1^{dc} \leftarrow t$ Downturn event end time

$t_0^{os} \leftarrow t + 1$ Downward OS event start time

else

if $p_h < p_t$ **then**

$p_h \leftarrow p_t$

$t_0^{dc} \leftarrow t$ Downturn event start time

$t_1^{os} \leftarrow t - 1$ Downward OS event end time

end

end

else

if $p_t \geq p_l * (1 + \theta)$ **then**

 event \leftarrow Upturn Event

$p_h \leftarrow p_t$

$t_1^{dc} \leftarrow t$ Upturn event end time

$t_0^{os} \leftarrow t + 1$ Upward OS event start time

else

if $p_l > p_t$ **then**

$p_l \leftarrow p_t$

$t_0^{dc} \leftarrow t$ Upturn event start time

$t_1^{os} \leftarrow t - 1$ Downward OS event end time

end

end

end

3.2 Long short-term memory (LSTM)

Recurrent Neural Networks (RNN), are a robust type of artificial neural network which process sequences by iterating through the sequence elements and maintaining a state containing information relative to previous states. Unlike the Feed-Forward neural networks, RNNs models can leverage the previous inputs' sequential information through memory gates. The RNNs memory, which is called recurrent hidden state, enable the network to predict the next item in the input data sequence. Practically, however, the length of the sequential information is limited to only a few steps back. Although RNNs should theoretically retain information from previous time-steps, such long-term dependencies are impossible to learn in practice. A common problem among RNNs is vanishing gradient when the gradients' information vanish while passing through a deep layered network. The gradient is the partial derivative of a function's output with respect to its inputs' changes. This problem

prevents the network from learning long-term dependencies which causes the learning process to slow down or stop altogether. Conversely, there is the exploding gradient problem in which the gradient’s information accumulate and result in a large gradient. In the “vanishing gradient” problem, the network assigns smaller values to the weight matrix, and in the “exploding gradient” problem, the opposite is true. As mentioned earlier, RNNs are not capable of learning long-term dependencies [11]. The LSTM models are an extension of RNNs and are designed to address the vanishing gradient problem. Generally, the LSTM model consists of three gates: forget, input, and output gates, as shown in Fig. 2. The forget gate is responsible for deciding to preserve or removing the existing information. The input gate determines the extent to which the new information will be added into the memory, and the output gate controls whether the current value in the cell contributes to the output [11].

- Forget Gate: In the forget gate block of the LSTM layer, the information from the current input x_t and the previous hidden state h_{t-1} is passed through an activation function (e.g. sigmoid). The gate output f_t will be a value between 0 and 1, where zero implies removing the learned value while one means to preserve the value. The output is computed as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

where b_f is called the bias value.

- Input Gate: This gate which determines the additions of new information to the LSTM memory has two layers. A sigmoid layer decides which values need to be updated and the hyperbolic tangent layer generates a vector of new values that will be added to the memory.

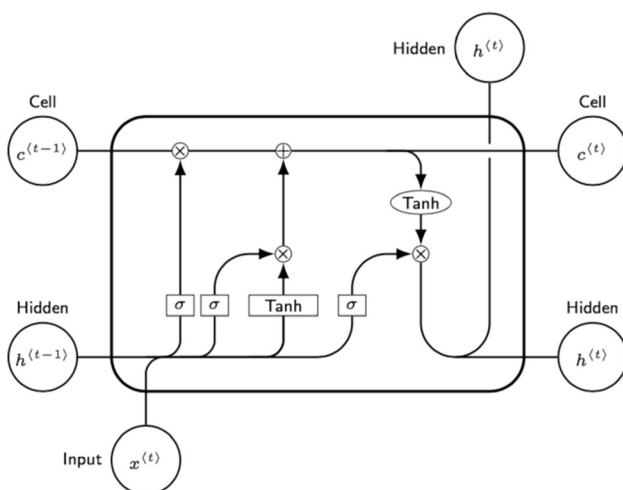


Fig. 2 LSTM architecture

The output value of the input gate is computed through the following formulas:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{5}$$

Together, these two layers update the LSTM memory, forgetting the current value by multiplying the old value and adding a new value $i_t * \tilde{C}_t$. The following represents its equation:

$$\tilde{C}_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

- Output Gate: Here the gate first uses a sigmoid function to determine which part of the LSTM memory contributes to the output. Subsequently, through the nonlinear tanh function, it maps the values between -1 and 1 .

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

Figure 2 is the depiction of the LSTM architecture.

3.3 Convolutional neural networks (CNN)

Convolutional Neural Network (CNN), designed by Lecun et al. [15] is a special type of Feed-Forward network with high performance in image processing and natural language processing [14]. The main parts of the CNN are the convolution and pooling layer. Each convolution layer contains different kernels. Following the convolutional operations, the high dimensional extracted features pass through a pooling layer to reduce the dimensionality.

$$l_t = \tanh(x_t * k_t + b_t) \tag{9}$$

In the above equation, l_t represents the convolution’s output, x_t is the input vector, k_t is the convolution kernel weights, and b_t is the bias. Although Convolutional Neural Network was initially designed for image processing, it can be utilised for time series forecasting. The reduced number of parameters by the CNN improves the efficiency of the model [23].

3.4 Support vector and random forest regression

Support Vector Machines proposed by Vapnik [33] formulate the binary classification problem as convex optimisation problems, which entails finding the maximum margin separating the hyperplane. Support vectors represent the optimal hyperplane. The introduction of an ϵ -insensitive region around the function forms epsilon-tube around the function, generalising the Support Vector Machine to Support Vector Regression. The so-called ϵ -

Table 1 Sample of tick price transformation into a 1000 Tick-Bars and DC confirmation points for GBPUSD

Open	High	Low	Close	Directional change
1.27605	1.27615	1.27250	1.27460	1.27615
1.27460	1.27605	1.27360	1.27400	1.27490
1.27400	1.27615	1.27400	1.27615	1.27435
1.27615	1.27615	1.27470	1.27555	1.27400
1.27565	1.27615	1.27380	1.27490	1.27515
1.27490	1.27555	1.27380	1.27435	1.27535
1.27380	1.27525	1.27355	1.27400	...
1.27475	1.27555	1.27400	1.27495	...
1.27500	1.27510	1.27470	1.27490	...
1.27495	1.27525	1.27460	1.27515	...
1.27515	1.27545	1.27500	1.27535	...

tube redefine the optimisation problem to find the tube with the best approximation of the continuous values function and balanced complexity and prediction error. Another widely used regression method in financial forecasting is Random Forest. With the intuition of combining multiple decision trees and a bootstrap aggregation technique, a Random Forest (RF) is an ensemble method in the field of classification and regression problems. Ensemble techniques employ multiple weak learners, e.g. decision trees, and create a strong one such as Random Forest. In Random Forest, the bootstrapping technique reduces the variance and maintains the low bias.

3.5 Data

Financial data comes in a variety of shapes and forms. The four essential financial data types are fundamental data, market data, analytics, and alternative data. To apply

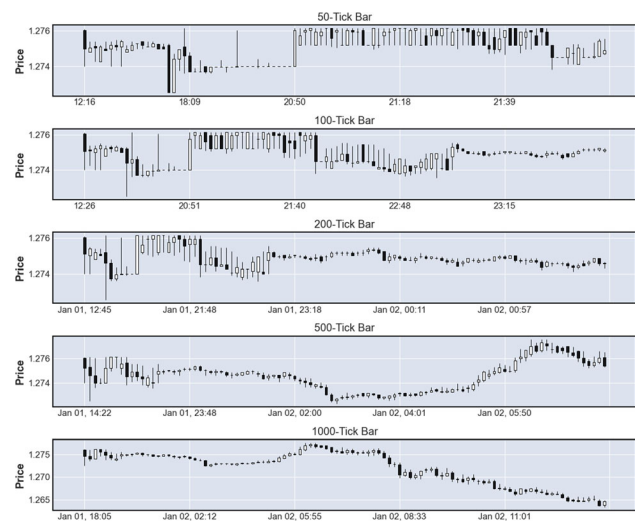


Fig. 3 The first 100 observations of GBPUSD with a predefined number of tick prices

Table 2 Durbin–Watson statistic of the currency pairs

Currency pair	50 tick-bar	100 tick-bar	200 tick-bar	500 tick-bar	1000 tick-bar
GBPUSD	2.294	2.163	2.126	2.094	2.092
EURUSD	2.591	2.447	2.320	2.180	2.131
USDCHF	2.289	2.181	2.158	2.138	2.112
USDCAD	2.678	2.511	2.379	2.386	2.461

Bold indicates tick bars with the lowest DW values

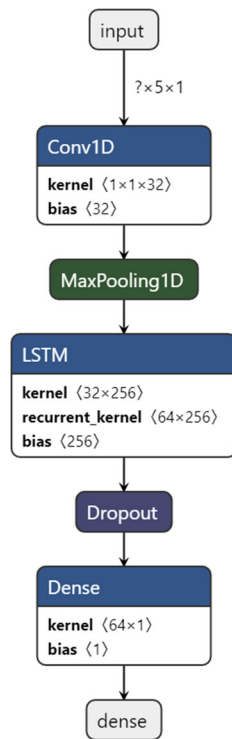
machine learning algorithms on unstructured financial data, we need to parse it and extract valuable information, then store those extractions in a regularized format. The tabular representations of data used in ML algorithms (i.e. table rows) equate to what finance practitioners refer to as bar in bar charts [7]. Time bars which perhaps are the most popular among market practitioners and academics are generated through sampling price information at fixed time intervals. The information usually includes; timestamp, volume-weighted average price, open, high, low, close, and traded volume. Time bars unrealistically process information at a fixed time interval, leading to an exhibition of poor statistical properties [7].

In financial jargon, a tick refers to a change in the price of a security from a trade to the next. In order to create tick bars, sample variables mentioned earlier will be extracted each time a predefined number of transactions occurs, allowing synchronising sampling with a proxy of information arrival. For instance, if we wish to generate 100-tick bars, we need to store the 100 price information and then extract the open, high, low, and close value from the observations. Mandelbrot and Taylor [18] found that sampling as the function of transaction numbers exhibit Gaussian distribution properties. In contrast, sampling over a fixed interval may follow a stable Paretian distribution, whose variance is infinite [7]. It should be mentioned that throughout this paper, tick bars and tick candles are used interchangeably. The sole difference between the two is that the tick candles are colour coded to reflect any increase or decrease in price.

3.6 Average true range

The average true range (ATR) is a technical analysis indicator that measures market volatility. It decomposes the whole range of an asset price for a specific period. It is typically derived from a moving average of length 14 of a series of true range values and can be calculated on an intra-day, daily, weekly or monthly basis. If the current high is above the prior period’s high and the low is below the prior period’s low (i.e. outside day) high less the low will be used as the True Range. In addition, in the case of a

Fig. 4 CNN-LSTM model



gap when the previous close is greater than the current high or the previous close is lower than the current low, or an inside day (i.e. when the current high is below the previous high and the current low is above the previous low), current high less the previous close or the current low less the previous close will be used. Following equations represents the calculation of ATR:

$$TR = \max[(H - L), |H - C_{previous}|, |L - C_{previous}|] \tag{10}$$

$$ATR = \frac{1}{n} \sum_i^n TR_i \tag{11}$$

$$ATR \% = \frac{ATR}{\text{current price}} \tag{12}$$

where TR_i is the true range, and n is the time period. In Eq. 12, $ATR\%$, is the ATR division by the current price of the asset. Table 1 illustrates a sample of raw tick prices transformed into tick bars, sampled for every one thousand observations. The open, high, low, and close are the first, highest, lowest, and last tick prices within a sequence of a thousand tick prices. The last column is the price at which the directional change occurs. The change in direction is confirmed if the price exceeds a threshold in either direction. The remaining values in the directional change column are excluded since no more $ATR\%$ -defined changes in direction happened in the sample.

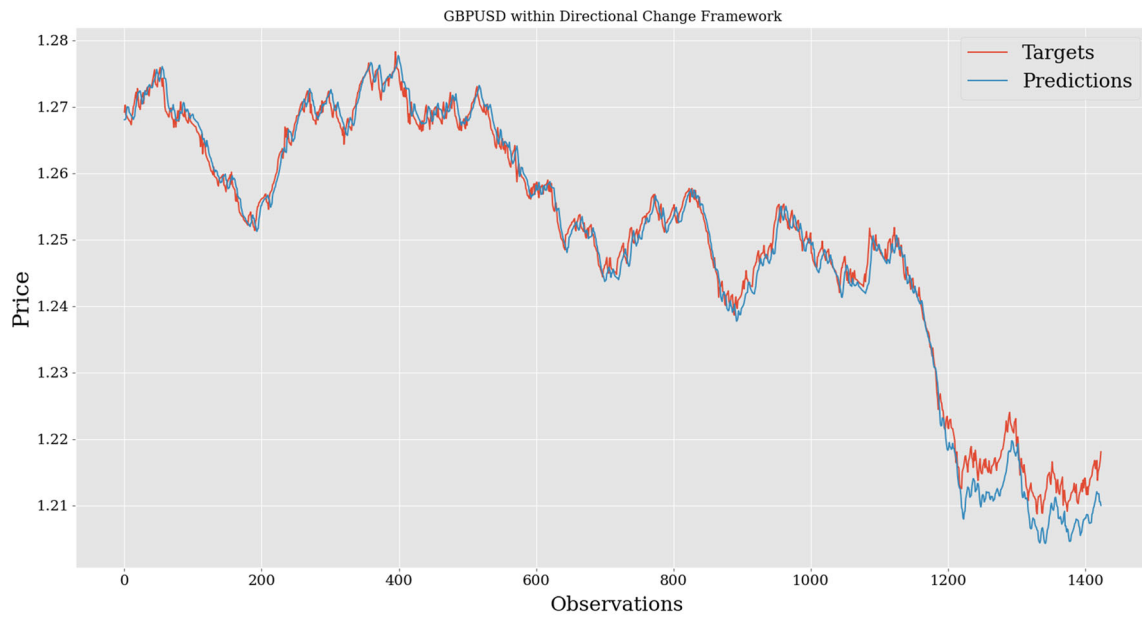
3.7 Experiment

This paper’s objective is to apply the CNN-LSTM network to the generated DC-based summaries of GBPUSD, EURUSD, USDCHF, and USDCAD tick prices to predict the following price of the directional change event. The initial dataset comprises of the currency pairs’ tick prices from January to August of 2019, in comma-separated variables (CSV) format. As we mentioned earlier, a tick price alludes to a change in an asset price from one trade to the next. Our model aims to predict the immediate step-ahead movement of the financial asset tick prices instead of the time prices. Note that predictions are short-term and sensitive to the threshold values, i.e., different user-defined thresholds produce different summaries of the price movements.

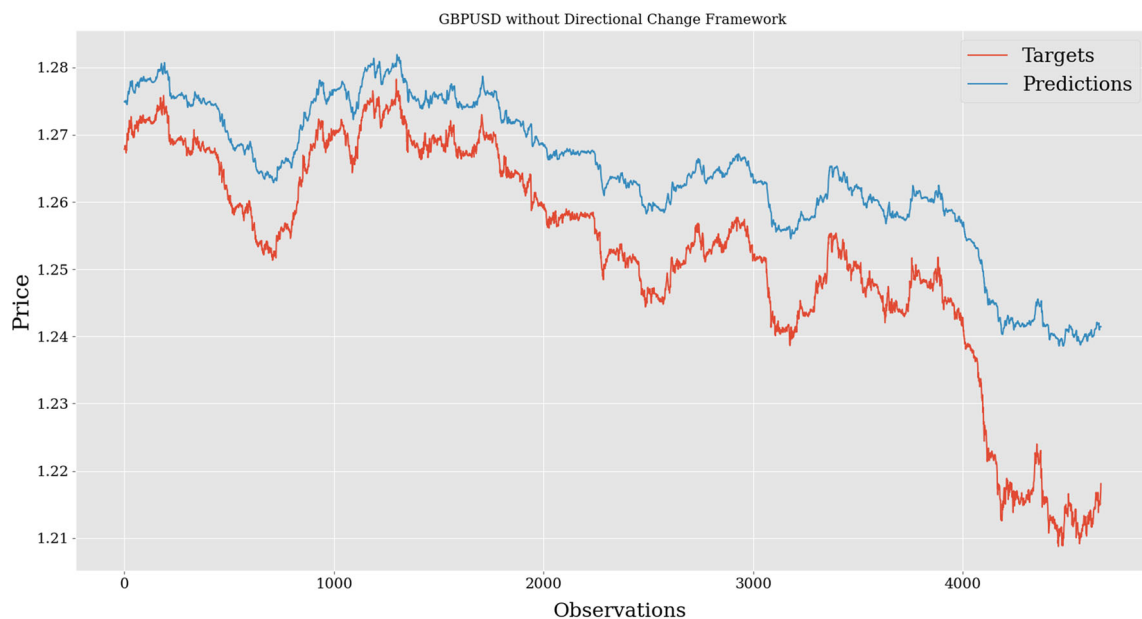
To generate the tick bars, we will aggregate 50, 100, 200, 500, 1000 data points from the original tick prices of the GBPUSD, EURUSD, USDCHF, USDCAD currency pairs. Every tick bar has an open, high, low, and close price. The open and close prices correspond to the price of the first and last trade. The high and close prices are the maximum and minimum prices within the range of the predefined number of ticks. Figure 3 is the depiction of the generated tick bars/candles from the GBPUSD tick prices with the predefined number of ticks. The tick bar with the

Table 3 CNN-LSTM parameters

Parameters	Value
Convolution layer filters	32
Convolution layer kernel size	1
Convolution layer padding	Same
Pooling layer pool size	3
Pooling layer padding	Same
Number of units in LSTM layer	64
LSTM activation function	<i>tanh</i>
Dropout rate	0.2
Optimizer learning rate	0.001



(a) CNN-LSTM prediction on GBPUSD DC summaries



(b) CNN-LSTM prediction on GBPUSD 1000 tick bars

Fig. 5 CNN-LSTM results within DC framework and on raw tick bars for GBPUSD

least auto-correlation will be used to generate the DC-based summaries. In order to obtain the least auto-correlated tick bar, the Durbin–Watson (DW) statistic was performed on all the currency pairs’ tick bars.

The DW test is calculated with the following formula:

$$DW = \frac{\sum_{t=2}^T (e_t - e_{t-1})^2}{\sum_{t=1}^T e_t^2} \quad (13)$$

The Durbin–Watson test reports a value from 0 to 4, where:

- $DW = 2$ is no auto-correlation.
- $0 < DW < 2$ is positive auto-correlation.
- $2 < DW < 4$ is negative auto-correlation.

Table 2 represents the Durbin–Watson results for the tick bars. As the results imply, 1000 tick-bar has the lowest DW value for GBPUSD, EURUSD, USDCHF and 200 tick-bar for the USDCAD pair. The Average True Range will be calculated for the tick-bars with the smallest DW and will then be used as the Directional Change threshold θ . As it

Table 4 Prediction accuracy results

	CNN-LSTM			Support vector regression			Random forest regression		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
(a) Results within DC framework									
GBPUSD	0.0142	0.0179	0.985	0.2909	0.3604	- 1.8945	0.2841	0.3543	- 1.7975
EURUSD	0.0188	0.0248	0.972	0.5746	0.6047	- 8.3019	0.5993	0.6294	- 9.0758
USDCHF	0.0301	0.0387	0.865	0.0917	0.1149	0.0035	0.1016	0.1220	- 0.1219
USDCAD	0.0182	0.0221	0.973	0.5791	0.5914	- 23.1734	0.5488	0.5579	- 30.1266
(b) Results without DC framework									
GBPUSD	0.0604	0.0697	0.359	0.3435	0.3942	- 3.1636	0.2501	0.3158	- 1.6722
EURUSD	0.0294	0.0368	0.946	0.5107	0.5409	- 6.9423	0.6003	0.6304	- 9.7862
USDCHF	0.0466	0.0516	0.772	0.1029	0.1291	- 0.0081	0.1133	0.1382	- 0.1549
USDCAD	0.0989	0.1094	0.548	0.5812	0.5948	- 21.071	0.5625	0.5725	- 28.2334

was previously mentioned, the Average True Range (ATR) is a market volatility measure and is typically calculated from the 14-day simple moving average of true range values. With the derived θ , DC-based summaries will be generated and used within a sliding window of length 5 to predict the next event value. The CNN-LSTM model, as its name implies, consists of a convolutional neural network layer and a long short-term memory layer. Figure 4 is the illustration of the employed model.

As demonstrated in Fig. 4, the convolutional layer outputs are passed into a max-pooling layer. In order to prevent the model from over-fitting, a dropout layer is placed following the LSTM layer. The number of Convolutional filters, LSTM units and activation function, as well as the Dropout percentage and optimizer learning rate, were determined through hyper-parameter tuning with KerasTuner [21]. Table 3 presents the parameters' setting for the CNN-LSTM model. The DC summaries of the currency pairs were divided into training, validation, and test sets, where 80% of data points constitute the training, and the remaining 20% is the test set. Moreover, 20% of the training set was used as the validation set to prevent data leakage. The training process was performed with the *Adam* optimiser and the mean squared error as the loss function. To evaluate the predictive performance of the model, the mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2) will be used. The followings are the equations for the MAE, RMSE, and R^2 (Table 1).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \tag{15}$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{16}$$

The CNN-LSTM model will be trained and validated with the DC summaries of GBPUSD, EURUSD, USDCHF, and USDCAD with an EarlyStopping of Keras callback API. Initially, DC summaries of the GBPUSD pair will be used to train and validate the model on the training and validation sets with respective 4,567 and 1,138 data points. Prediction on the test set, which is considered the out-of-sample set, resulted in a 0.0142 mean absolute error and a 0.0179 root mean squared error. Figure 5a represents the prediction of the model on the GBPUSD DC summaries. As it is observable, the model has reached a reasonably well prediction throughout the summaries with the coefficient of determination of 0.985. The accuracy of prediction has dwindled near the end of the graph. To explore the predictive capability of the CNN-LSTM model within the directional change framework and on the raw tick bars, we applied the identical CNN-LSTM model on the close price of the 1000 tick bar dataset. Training and validating the CNN-LSTM model on the GBPUSD raw 1000 tick bar dataset with the respective number of 14,921 and 3727 observations resulted in 0.0604 mean-absolute error (MAE), and 0.0697 root mean squared error (RMSE). We then utilised the trained model to perform predictions on the out-of-sample dataset. From Table 4b, in the absence of the DC Framework, the coefficient of determination has plummeted from 0.985 to 0.359. Figure 5b portrays this noticeable decline in the prediction accuracy of the model. The same steps were applied for EURUSD, USDCHF, and USDCAD currency pairs. With the suggestion of Table 4 and the comparison of Fig. 6a and b, an increase in the MAE and RMSE metrics from 0.0188 to 0.0294 and 0.0248 to 0.0368 is discernible. Furthermore, the coefficient of determination (R^2) for EURUSD has decreased



Fig. 6 CNN-LSTM results within DC framework and on raw tick bars for EURUSD

from 0.972 to 0.946. Despite capturing the overall trend of the USDCHF, distinguished from Fig. 7a and b, metrics altogether corroborate the substantial drop in the accuracy of the CNN-LSTM model. Both MAE and RMSE have risen from 0.0301 to 0.0466 and from 0.0387 to 0.0516. The R^2 has declined from 0.865 to 0.772. Figure 8a substantiates the prediction accuracy of the CNN-LSTM model within the DC framework. The model captured the overall trend correctly and predicted more than 6000 observations with the coefficient of determination (R^2) of

0.973. In Fig. 8b the performance of the model in predicting nearly three times more observations without DC framework plummeted to 0.548. For the USDCAD, MAE and RMSE have surged from 0.0182 to 0.0989 and from 0.0221 to 0.1094. R^2 has plunged from 0.973 to 0.548. We observed that the CNN-LSTM model, within the DC framework, outperforms itself with a considerable margin. Consequently, applying the CNN-LSTM model within the DC framework for the GBPUSD, EURUSD, USDCHF, and USDCAD currency pairs enhances the accuracy of the



Fig. 7 CNN-LSTM results within DC framework and on raw tick bars for USDCHF

prediction in all performance metrics. It is concluded from the results that applying the CNN-LSTM architecture within the directional change framework improves the accuracy of prediction for high-frequency FX data. Support Vector and Random Forest regression, two widely used machine learning techniques in financial forecasting, were also utilised to compare to the CNN-LSTM model. Both models' hyper-parameters were tuned with RandomisedSearchCV [22] and used in the same fashion as the CNN-LSTM with and without DC framework. It is concluded from Table 4 that Support Vector, and Random

Forest regression failed to perform an acceptable prediction with significantly high error and negative coefficient of determination (R^2).

Summarily, the tick bars were created from raw tick prices and the least auto-correlated were determined using the Durbin-Watson statistic. Next, the least auto-correlated tick bars were used to calculate the ATR value, which then was used as the Directional Change threshold θ . Then, the DC summaries of the tick bars were generated. Finally, the proposed model was applied to the mentioned DC summaries of all the currency pairs as well as their raw tick

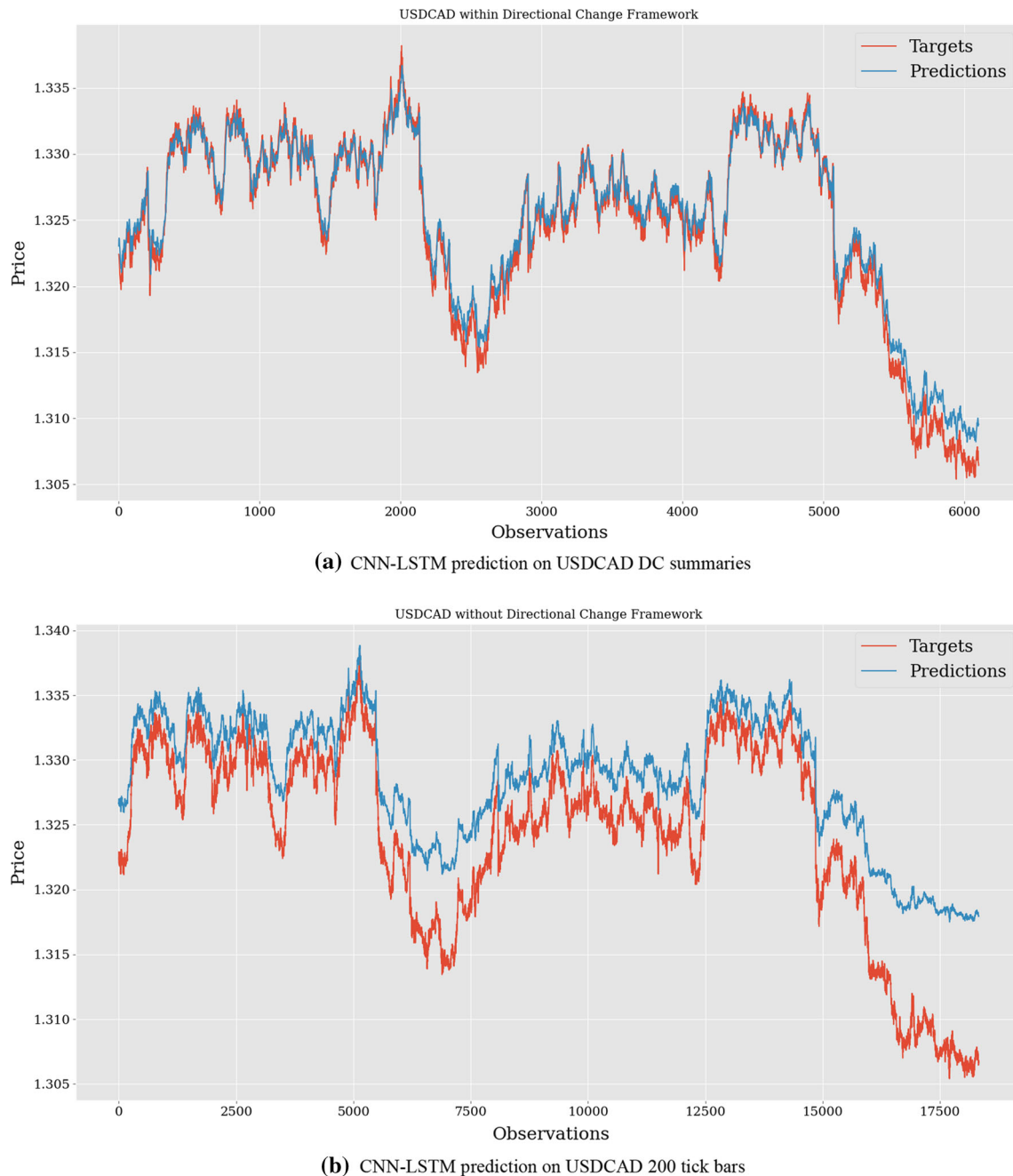


Fig. 8 CNN-LSTM results within DC framework and on raw tick bars for USDCAD

bars to investigate the performance of the CNN-LSTM model with and without the DC framework.

4 Conclusions and future work

This paper has investigated applying the CNN-LSTM model within the Directional Change (DC) framework, an approach to summarise price movement by transforming a time series price curve into an intrinsic time curve to

predict the subsequent event price. An event is identified by a significant change in the price of an asset, defined as a price change greater than a predefined threshold value θ . The threshold θ is determined with the Average True Range (ATR) indicator. The CNN-LSTM employs the DC summaries of tick bars with the lowest Durbin-Watson statistic for GBPUSD, EURUSD, USDCHF, and USDCAD currency pairs as the model's input. The same model was applied to the closing prices of the currency pairs tick bars without the DC framework to inspect the model's

performance. The experimental results suggest that the CNN-LSTM performance improves significantly within the directional change framework concerning MAE, RMSE, and R^2 metrics for all the currency pairs.

In future research, we intend to apply our model to predict more extended periods and experiment with more complex GRU and BiLSTM architectures on different currency pairs and financial assets. Due to the fact that thresholds are determined based on the practitioner's preferences, it would be of importance and interest to explore ways to determine the Directional Change threshold dynamically to address the sensitivity of the model to thresholds.

Data availability statement The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

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. Abu Hammad AA, Alhaj Ali SM, Hall EL (2007) Forecasting the Jordanian stock prices using artificial neural network. In: Intelligent engineering systems through artificial neural networks, vol 17. ASME Press. <https://doi.org/10.1115/1.802655.paper42>
2. Bakhach A, Tsang E, Ng WL, Chinthalapati VLR (2016) Backlash agent: a trading strategy based on directional change. In: 2016 IEEE symposium series on computational intelligence (SSCI), pp 1–9. <https://doi.org/10.1109/SSCI.2016.7850004>
3. Chen J, Tsang E (2020) Detecting regime change in computational finance data science, machine learning and algorithmic trading, 1st edn. CRC Press, Boca Raton
4. Choudhury S, Ghosh S, Bhattacharya A, Fernandes KJ, Tiwari MK (2014) A real time clustering and svm based price-volatility prediction for optimal trading strategy. Neurocomputing 131:419–426
5. Fischer TG, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. Eur J Oper Res 270:654–669
6. Di Persio L, Honchar O (2017) Recurrent neural networks approach to the financial forecast of Google assets. Int J Math Comput Simul 11:7–13
7. de Prado ML (2018) Advances in financial machine learning. Wiley, New York
8. Glattfelder J, Dupuis A, Olsen R (2011) Patterns in high-frequency FX data: discovery of 12 empirical scaling laws. Quant Finance 11:599–614
9. Golub A, Glattfelder JB, Olsen RB (2017) The alpha engine: designing an automated trading algorithm. Innov Meas Indic eJ
10. Guillaume DM, Dacorogna M, Davé RR, Müller UA, Olsen R, Pictet O (1997) From the bird's eye to the microscope: a survey of new stylized facts of the intra-daily foreign exchange markets. Finance Stoch 1:95–129. <https://doi.org/10.1007/s007800050018>
11. Hochreiter S, Schmidhuber J (1996) LSTM can solve hard long time lag problems. In: NIPS
12. Hu Y (2018) Stock market timing model based on convolutional neural network—a case study of Shanghai composite index. Finance Econ 4:71–74
13. Karmiani D, Kazi R, Nambisan A, Shah A, Kamble V (2019) Comparison of predictive algorithms: backpropagation, SVM, LSTM and Kalman filter for stock market. In: 2019 amity international conference on artificial intelligence (AICAI), pp 228–234. <https://doi.org/10.1109/AICAI.2019.8701258>
14. Kim BS, Kim T (2019) Cooperation of simulation and data model for performance analysis of complex systems. Int J Simul Model 18:608–619. [https://doi.org/10.2507/IJSIMM18\(4\)491](https://doi.org/10.2507/IJSIMM18(4)491)
15. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324. <https://doi.org/10.1109/5.726791>
16. Li J, Pan S, Huang L, Zhu X (2019) A machine learning based method for customer behavior prediction. Teh Vjesn Tech Gaz 26:1670–1676
17. Li Y, Ma W (2010) Applications of artificial neural networks in financial economics: a survey. In: 2010 international symposium on computational intelligence and design, vol 1, pp 211–214
18. Mandelbrot B, Taylor HM (1967) On the distribution of stock price differences. Oper Res 15(6):1057–1062
19. Mehtab S, Sen J (2020) Stock price prediction using convolutional neural networks on a multivariate timeseries
20. Nelson D, Pereira A, de Oliveira R (2017) Stock market's price movement prediction with LSTM neural networks, pp 1419–1426. <https://doi.org/10.1109/IJCNN.2017.7966019>
21. O'Malley T, Bursztein E, Long J, Chollet F, Jin H, Invernizzi L et al (2019) Kerastuner. <https://github.com/keras-team/keras-tuner>
22. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830
23. Qin L, Yu N, Zhao D (2018) Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video. Teh Vjesn 25:528–535
24. Roondiwala M, Patel H, Varma S (2017) Predicting stock prices using LSTM. IntJ Sci Res 6:1754–1756
25. Saad E, Prokhorov D, Wunsch D (1998) Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEEE Trans Neural Netw 9(6):1456–1470. <https://doi.org/10.1109/72.728395>
26. Salis VE, Kumari A, Singh A (2019) Prediction of gold stock market using hybrid approach. Int J Eng Res Technol 8:803–812
27. Sen J (2018) Stock price prediction using machine learning and deep learning frameworks
28. Sezer OB, Ozbayoglu AM (2018) Algorithmic financial trading with deep convolutional neural networks: time series to image

- conversion approach. *Appl Soft Comput* 70:525–538. <https://doi.org/10.1016/j.asoc.2018.04.024>
29. Shen S, Jiang H, Zhang T (2012) Stock market forecasting using machine learning algorithms. Stanford University, Stanford
 30. Tkáč M, Verner R (2016) Artificial neural networks in business: two decades of research. *Appl Soft Comput* 38:788–804
 31. Tsang E, Chen J (2018) Regime change detection using directional change indicators in the foreign exchange market to chart Brexit. *IEEE Trans Emerg Top Comput Intell* 2(3):185–193. <https://doi.org/10.1109/TETCI.2017.2775235>
 32. Tsang E, Tao R, Serguieva A, Ma S (2017) Profiling high-frequency equity price movements in directional changes. *Quant Finance* 17:217–225
 33. Vapnik VN (2000) The nature of statistical learning theory. In: *Statistics for engineering and information science*
 34. Wang JJ, Wang JZ, Zhang ZG, Guo SP (2012) Stock index forecasting based on a hybrid model. *Omega* 40(6):758–766
 35. Wang P, Lou Y, Lei L (2017) Research on stock price prediction based on BP wavelet neural network with mexican Hat wavelet basis. Atlantis Press, Amsterdam, pp 99–102. <https://doi.org/10.2991/iceemr-17.2017.25>
 36. White (1988) Economic prediction using neural networks: the case of IBM daily stock returns. In: *IEEE 1988 international conference on neural networks*, vol 2, pp 451–458. <https://doi.org/10.1109/ICNN.1988.23959>
 37. Zhang D, Zhou L (2004) Discovering golden nuggets: data mining in financial application. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 34(4):513–522
 38. Zbikowski K (2015) Using volume weighted support vector machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Syst Appl* 42:1797–1805
 39. Zhang G (2003) Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* 50:159–175
 40. Zhang L, Wang F, Xu B, Chi W, Wang Q, Sun T (2018) Prediction of stock prices based on LM-BP neural network and the estimation of overfitting point by RDCI. *Neural Comput Appl* 30:1425–1444. <https://doi.org/10.1007/s00521-017-3296-x>
 41. Zhuge Q, Xu L, Zhang G (2017) LSTM neural network with emotional analysis for prediction of stock price. *Eng Lett* 25:167–175

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