



# Applying attention-based BiLSTM and technical indicators in the design and performance analysis of stock trading strategies

Ming-Che Lee<sup>1</sup> · Jia-Wei Chang<sup>2</sup> · Sheng-Cheng Yeh<sup>1</sup> · Tsorng-Lin Chia<sup>1</sup> · Jie-Shan Liao<sup>1</sup> · Xu-Ming Chen<sup>1</sup>

Received: 1 July 2021 / Accepted: 4 December 2021 / Published online: 28 January 2022  
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

## Abstract

With the development of the Internet, information on the stock market has gradually become transparent, and stock information is easy to obtain. For investors, investment performance depends on the amount of capital and effective trading strategies. The analysis tool commonly used by investors and securities analysts is technical analysis (TA). Technical analysis is the study of past and current financial market information, and a large amount of statistical data is used to predict price trends and determine trading strategies. Technical indicators (TIs) are a type of technical analysis that summarizes possible future trends of stock prices based on historical statistical data to assist investors in making decisions. The stock price trend is a typical time series data with special characteristics such as trend, seasonality, and periodicity. In recent years, time series deep neural networks (DNNs) have demonstrated their powerful performance in machine translation, speech processing, and natural language processing fields. This research proposes the concept of attention-based BiLSTM (AttBiLSTM) applied to trading strategy design and verified the effectiveness of a variety of TIs, including stochastic oscillator, RSI, BIAS, W%R, and MACD. This research also proposes two trading strategies that suitable for DNN, combining with TIs and verifying their effectiveness. The main contributions of this research are as follows: (1) As our best knowledge, this is the first research to propose the concept of applying TIs to the LSTM-attention time series model for stock price prediction. (2) This study introduces five well-known TIs, which reached a maximum of 68.83% in the accuracy of stock trend prediction. (3) This research introduces the concept of exporting the probability of the deep model to the trading strategy. On the backtest of TPE0050, the experimental results reached the highest return on investment of 42.74%. (4) This research concludes from an empirical point of view that technical analysis combined with time series deep neural network has significant effects in stock price prediction and return on investment.

**Keywords** Attention mechanism · Long short-term memory · Stock price trend forecast · Trading strategies

## 1 Introduction

In the stock market, different investors have different opinions on whether the stock price can be predicted. Cootner [1] believes that in the efficient market, the stock price has reflected all public information at any time;

therefore, the stock price is unpredictable. Some scholars [2] believe that the best strategy for stock investment is to buy and hold for a long time to earn dividend income. The methods of financial commodity analysis can be divided into fundamental analysis and technical analysis [3, 4]. The argument of the former is that the value of financial products is determined by the overall economic environment, the industrial environment to which it belongs, and individual companies and operating performance. On the other hand, there is a lot of information in the stock market for investors to use and refer to, including daily stock price changes, trading volume, stock price fluctuations, changes in chips, average costs, margin trading, etc. The information with reference value has been aggregated and

---

✉ Jia-Wei Chang  
jwchang@nutc.edu.tw

<sup>1</sup> Department of Computer and Communication Engineering, Ming Chuan University, Taoyuan City, Taiwan

<sup>2</sup> Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung City, Taiwan

analyzed, and expressed by certain data special formulas, the so-called technical indicators (TIs) [5]. Technical indicators are sufficient to simplify market information and reflect on values or graphs for investors to refer to and formulate trading strategies.

In recent years, deep neural network [6] and various deep learning algorithms have shined in major pattern recognition and machine learning competitions. The vigorous development of deep neural networks has not only created a new field of machine learning research, but its various applications have gradually appeared in people's lives, such as speech recognition, emotion recognition, natural language processing, and image recognition [7–12]. In the field of deep learning, recurrent neural network (RNN) [13], long short-term memory (LSTM) [14], and attention mechanism [15, 16] are particularly good at processing time series data, such as natural language processing, machine translation, speech recognition, and financial index prediction [17–19].

The application of time series neural networks to financial commodity trends and price forecasts has gradually become the main trend of financial technology (FinTech) recently [20]. Nelson et al. [21] first proposed the concept of using LSTM neural networks to predict the future trend of stocks. The model label is a binary classification: compared with the closing price of the previous day, up or down. The experiment target was selected by four stocks from the Brazilian stock exchange, and the result reached a maximum of 55.9% forecast accuracy.

Chen et al. [22] proposed an RNN-boost method and used latent Dirichlet allocation (LDA) to select data features. The model was applied to Shanghai-Shenzhen 300 Stock Index (HS300) index prediction. The results show that the time series model is better than other traditional methods, such as support vector regression (SVR)-based approach [23].

Li et al. [24] first proposed the MI-LSTM (Multi-Input) prediction model based on attention mechanism. In the proposed model, attention layer is followed by a LSTM layer, and the prediction results are obtained through Softmax. The experimental target is Shanghai-Shenzhen CSI 300 comprehensive index, and the results show that sequence model with the attention mechanism is better than general sequence models.

Li et al. [25] proposed a Markov random fields combined with a multitask RNN model and applied it to the stock price movement prediction. The model reached 67.97, 66.80, and 68.95% prediction accuracy in the Chinese Securities Index (CSI) 200/300/500.

Livieris et al. [26] proposed a CNN-LSTM-based model to predict the international gold price trend. In their proposed model, the input sequence is first converted to a  $32 \times 64$  convolution operation and then input to the

sequence LSTM to obtain the final prediction result. The experimental results are slightly better than SVR kernel method.

Hu [27] proposed the concept of using complete ensemble empirical mode decomposition to remove noise, weighted oil news, and financial news with TF-IDF as input attributes, combined with attention-LSTM to predict oil prices. The experimental results show that the proposed method is significantly better than SVR, AdaBoost, random forest, and other methods in the RMSE evaluation.

Based on the above literature, most studies have not conducted in-depth discussions on the application of technical indicators to sequential deep networks. In our previous research [28], we explored the effectiveness and practicality of various financial analysis technical indicators in time series deep learning networks. In financial commodity trading, it is common to analyze and discuss prices at various levels, such as fundamental, technical, and chip analysis. Technical analysis is the most frequently cited as the primary tool for smart stock selection by various trading software. This research explores the possibility of using technical analysis as the feature selection of deep networks. This study designed feature selection methods for famous technical indicators, such as 9 K-9D, 16 K-16D, RSI, MACD, and Williams%R, and measured it with a four-layer LSTM. The main contribution of [28] is to define the regularization of mainstream technical indicators applied to sequential neural networks.

In view of the above literature, the recent related research of deep learning applied to FinTech mainly focuses on the rise and fall forecast of financial products and model design. Research on the introduction of TAs into DNN models is very rare. The subsequent in-depth discussion on trading strategy research is even more lacking. At present, the research of financial technical indicators and deep learning are two independent research directions. How to effectively combine the two research fields and apply them to the design of trading strategies, and even automated trading based on deep learning, is a major challenge at the practical application level. This research proposes an attention-based BiLSTM neural network combined with trading strategy. In this study we analyze the performance of widely used TIs, including KD, RSI, BIAS, MACD, and Williams%R on the time series neural network. This research also demonstrates the effectiveness of using time series neural networks to construct financial commodity trading strategies. The remainder of this paper is as follows: Sect. 2 is the literature and techniques review; Sect. 3 is the methodology of this research; Sect. 4 is the experimental design, results and discussion, and the last section is conclusion and future work.

## 2 Related techniques and literature review

### 2.1 Machine learning and deep learning

Machine learning is a main branch of artificial intelligence (AI). AI generally refers to the reasoning, induction, and knowledge gained through programs. Machine learning is the mathematical rules and means to realize artificial intelligence [29, 30].

Machine learning and deep learning are widely known to the public after AlphaGo developed by DeepMind (acquired by Google in 2014) defeated the world champion of Go [31]. In 2018, the “*Turing Award*” was awarded to well-known deep learning scholars, which pushed machine learning/deep learning to the peak in one fell swoop and left an extremely important milestone in history [32]. Deep learning technologies have been widely used in different fields in recent years, such as autonomous driving systems [33], voice recognition systems [34, 35], AIoT [36, 37], face recognition [38], smart home and mart city [39–42], smart campus [43, 44], machine translation [45–47], image processing and retrieval [48–50], natural language processing [51, 52], etc.

The learning rule of the neural network is mainly to give an error evaluation function  $\mathbb{E}$  with  $N$  training samples and obtain the updated weight by the way of gradient-derived transmission by delta rules [53–55]. Given the weight  $w_t$  and bias  $b_t$  of  $t$ -layer, the gradient update method is as follows, where the learning rate  $\gamma$  is a positive number.

$$w_{t+1} = w_t - \gamma_t \nabla_w \mathbb{E} \tag{1}$$

$$b_{t+1} = b_t - \gamma_t \nabla_b \mathbb{E} \tag{2}$$

The error term of the  $(l + 1)$ th layer can be obtained from the weight of the previous layer, and weight and bias update are calculated as follows:

$$\delta^l = (\mathbf{W}^{l+1})^T \delta^{l+1} \circ f'(\mathbf{u}^l) \tag{3}$$

$$\nabla_{\mathbf{W}^{(l)}} \mathbb{E} = \mathbf{x}^{l-1} (\delta^l)^T \tag{4}$$

$$\nabla_{\mathbf{b}^{(l)}} \mathbb{E} = \delta^l \tag{5}$$

where  $\circ$  represents element-wise multiplication,  $f(\bullet)$  is the nonlinear transformation equation in a neuron,  $\mathbf{W}^{l+1}$  denotes the weight matrix from  $l$ -layer to  $(l + 1)$ -layer.

### 2.2 Recurrent Neural Network (RNN)

In the fundamental neural network, the training and discrimination of individual data often operate independently, which is relatively unsuitable for serial data input such as audio or natural language. For time series data with extremely high correlation between the context, it is relatively suitable to use recurrent neural network (RNN) [13].

If a fully connected neural network or a convolutional neural network is used to process the same data, the correlation between the data may be ignored and the prediction may be inaccurate.

The structure of the RNN is shown in Fig. 1. When the training data is input, the hidden layer will be synchronously transferred to the next hidden layer, to maintain the dependence between data. In the above architecture,  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\mathbf{W}$  are shared weights. There will be a nonlinear transformation from  $\mathbf{U}$  to the hidden layer, usually using hyperbolic tangent function ( $\tanh$ ). In addition, the multi-category output is converted through Softmax Function before  $\mathbf{V}$  to output. The RNN architecture generally uses the cross-entropy error function.

### 2.3 Long Short-Term Memory (LSTM)

The structure of long short-term memory (LSTM) cell is shown in Fig. 2 [14]. The difference between LSTM and the basic type of RNN is that the memory of RNN will decrease as the data sequence increases. From a theoretical point of view, the gradient feedback of the hidden layer will decrease layer by layer as the sequence data increases, and LSTM can effectively improve this problem. With its advantages, LSTM has been widely used in various time series data and applications, such as pedestrian trajectory prediction [56], chemistry and material science [57], and COVID-19 trend prediction [58].

LSTM cells are composed of input gates, forget gates, output gates, and unit states. The computation of hidden state  $h_t$  in a specific LSTM cell is as follows:

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{6}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{7}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{8}$$

$$O_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{9}$$

$$C_t = i_t \circ \tilde{C}_t + f_t \circ C_{t-1} \tag{10}$$

$$h_t = O_t \circ \tanh(C_t) \tag{11}$$

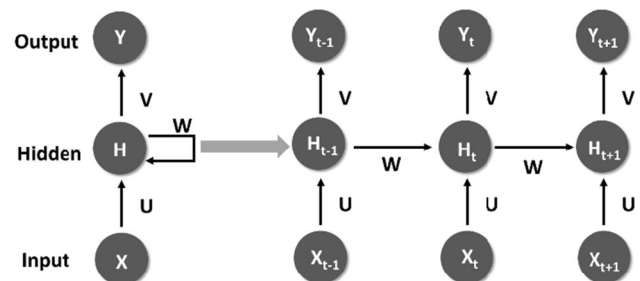


Fig. 1 Recurrent neural network architecture

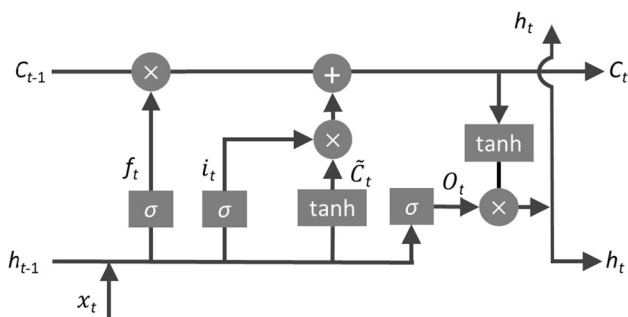


Fig. 2 A long short-term memory cell

where  $\sigma(\cdot)$  is the sigmoid activation function;  $\circ$  represents element-wise product;  $i_t, f_t$ , and  $O_t$ , indicate the input, forget, and output gates, respectively.  $C_t$  is the activation vectors, and  $\tilde{C}_t$  represents the intermediate vector of cell state. In LSTM model, each LSTM neural unit will go through a process of  $C_{t-1} \rightarrow C_t$ ,  $C_t$  is the current memory content, and  $C_{t-1}$  is the memory content of the previous moment. The content of time  $t - 1$  will be updated or deleted at time  $t$ . The forget gate uses the important measurement of past memory and the comprehensive calculation of  $C_{t-1}$  to play the role of memory screening. The input importance factor is adjusted by the tanh function and then added to  $C_{t-1}$  to expand the memory capacity.

2.4 Attention mechanism

Attention mechanism was first proposed in the field of computer vision, and then the Google Mind team applied the attention mechanism to the RNN model for image classification and achieved extremely high accuracy [59]. Subsequently, Bahdanau and Bengio [60] applied attention mechanism to synchronous machine translation tasks, which is a successful application on NLP. In 2017, the Google Brain team published <Attention Is All You Need> [16] to direct the attention mechanism to an independent structure instead of being attached to the existing CNN/RNN architecture.

The attention mechanism can simulate that when humans are watching a certain image or understanding things, they do not look at the whole picture in a full field of view, but instead focus their attention on a specific area. Attention mechanism is an improved version of the sequence-to-sequence (Seq2Seq) model in RNN [61]. In the sequence-to-sequence encoding stage, the hidden state of each time state is passed forward, and finally a set of vectors of uniform dimensions are generated before decoding. Seq2Seq causes each training data to correspond to the same set of context vectors. The attention mechanism (Fig. 3) is aimed at improving this situation. It passes the hidden state of each training to the back-end decoder,

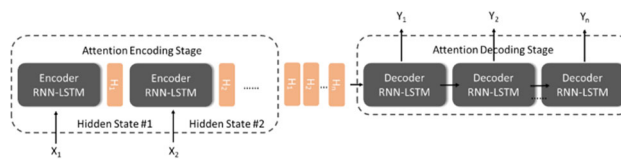


Fig. 3 Attention mechanism with RNN-LSTM

so that the important state in the timing will be learned by the network (i.e., the focus of attention).

3 The proposed attention-based BiLSTM stock trading strategy

3.1 Technical Indicator preprocessing

The technical indicators imported in this research include stochastic oscillator (KD), relative strength index (RSI), bias ratio (BIAS), William’s oscillator (W%R), and moving average convergence/divergence (MACD). Since the range of each technical indicator value is different from the meaning of stock market momentum, this study adopts the concept of Lee et al. [28] and normalizes the technical indicators as the following:

- i. Stochastic Oscillator (KD):

$$\begin{cases} A : K_{t-1} < D_{t-1} \text{ AND } D_t < K_t, \\ \text{AND}(\text{one of } K_t, K_{t-1}, \text{ and } K_{t-2} \text{ is } 20 \sim 80) \\ B : K_{t-1} > D_{t-1} \text{ AND } D_t > K_t, \\ \text{AND}(\text{one of } K_t, K_{t-1}, \text{ and } K_{t-2} \text{ is } 20 \sim 80) \\ C : \text{others} \end{cases} \tag{12}$$

where  $K_t, D_t$  are the  $K$ - $D$  values on the trading day  $t$  [2]. When the  $K$  line is higher than the  $D$  line but the  $K$  falls below the  $D$  value in the next day, it is called a death cross. When the  $K$  line is lower than the  $D$  line but the  $K$  breaks through the  $D$  value in the next day, it is a golden cross.

- ii. Relative Strength Index (RSI)

$$\begin{cases} A : RSI_t \leq 20 \\ B : 20 < RSI_t < 80 \\ C : RSI_t \geq 80 \end{cases} \tag{13}$$

where RSI value is close to 80 and 20 and is the time point for the trend reversal [62]. A value close to or greater than 80 indicates a downtrend, and a value close to or less than 20 implies a high possibility of rising.

- iii. Bias Ratio (BIAS)

BIAS [63] is the deviation rate of the moving average, and its value may be positive or negative. The theoretical cutoff point is between + 3 and -3%.

A positive deviation means that the upward momentum is large, and a negative deviation is the opposite. The normalization of BIAS input into the deep network in this study is as follows:

$$\begin{cases} A : \text{BIAS}_t \leq -0.03 \\ B : -0.03 < \text{BIAS}_t < 0.03 \\ C : \text{BIAS}_t \geq 0.03 \end{cases} \quad (14)$$

iv. William’s Oscillator (W%R)

William’s oscillator [64] is an oscillatory indicator that measures the overbought and oversold conditions of stocks. Its range and reversal signal are similar to the concept of the RSI indicator. The normalization of W%R input into the deep network in this study is as follows:

$$\begin{cases} A : \text{W\%R}_t \geq 80 \\ B : 20 < \text{W\%R}_t < 80 \\ C : \text{W\%R}_t \leq 20 \end{cases} \quad (15)$$

v. Moving Average Convergence/Divergence (MACD)

MACD [65] is used to evaluate the short-term and long-term cross-state of the moving average. Its value can also be positive or negative. When the negative value turns positive, it means that the short-term upward trend breaks through the long-term stalemate state, and vice versa. The normalization of MACD input into the deep network in this study is as follows, where DIF is the difference between the short-term trend and the long-term trend:

$$\begin{cases} A : \text{DIF}_{t-1} < 9\text{MACD}_{t-1} \text{ and } 9\text{MACD}_t < \text{DIF}_t \\ B : \text{DIF}_{t-1} > 9\text{MACD}_{t-1} \text{ and } 9\text{MACD}_t > \text{DIF}_t \\ C : \text{others} \end{cases} \quad (16)$$

### 3.2 System architecture

The overall framework of the proposed AttBiLSTM model and trading strategy design is shown in Fig. 4. The system first normalizes KD, RSI, BIAS, W%R, and MACD with the methods mentioned in Sect. 3. The model uses stock

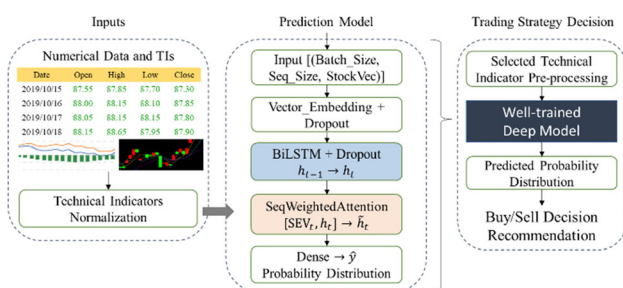


Fig. 4 The overall framework of the proposed AttBiLSTM prediction model applied to trading strategy design

daily trading information, including opening price, closing price, highest price, lowest price, and trading volume plus normalized technical indicators to form a vector (i.e., StockVec) for training. After preprocessing, StockVec will be input into the well-trained model. The model uses the LSTM + attention architecture to predict and output the probability distribution of the predicted categories. The final predicted distribution will be provided to users as a reference for buying and selling decisions.

The operation procedure of the proposed deep model is shown in Fig. 5. The LSTM architecture used in this study is BiLSTM, which is more sensitive to the dependence between stock prices and technical indicators in a time interval. The proposed BiLSTM algorithm is shown in Algorithm 1. The model input is a 3D tensor, including StockVec, sequence length and batch size. The daily changing technical indicators and basic data form a set of TI embedding and then form a BiLSTM sequence input according to the time series determined by the experiment. As shown in Fig. 3 of Sect. 3, BiLSTM is used as the encoder stage of the model, the operation is to learn the weight dependence of the StockVec of the input sequence in both forward and backward directions. While the BiLSTM operation is completed, different time series data will produce corresponding context vector via the attention mechanism in the given batch.

After the decoding vector is generated through the previous module, the probability of rising, falling or flat vector is obtained through the dense layer and the Softmax function [30], which predicts the probability for the  $j$ th class from input vector  $\mathbf{X}$ :

$$P(\text{output} = j|\mathbf{X}) = \frac{e^{\mathbf{X}^T w_j}}{\sum_{s=1}^S e^{\mathbf{X}^T w_s}} \quad (17)$$

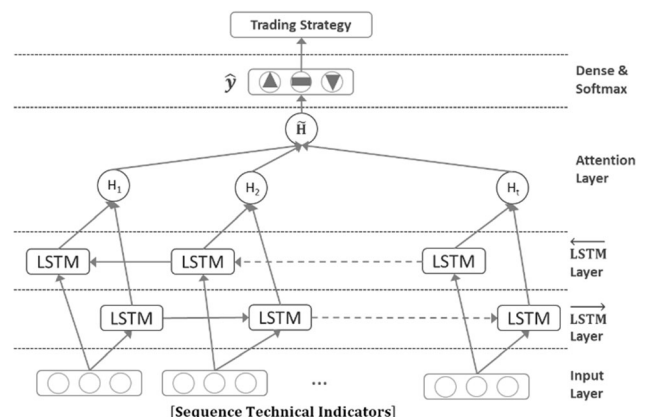


Fig. 5 The proposed AttBiLSTM operation procedure

**Algorithm 1.** BiLSTM-Attention Training

```

Inputs:  $\mathbb{D} = \{(X_n, Y_n); n = 1, \dots, N\}$ ;
            $X = (x_1, x_2, \dots, x_s)^T \in \mathbb{R}^s$ 
            $Y = (y_1, y_2, \dots, y_s)^T \in \mathbb{R}^s$ 
           }
           Maximum training epoch Epoch
           Sequence Length Seq
Output: Well-trained Model M
1  foreach  $(x_n, y_n) \in \mathbb{D}$ 
2  |   Normalize the technical indicators in  $x_n$ , convert data sequence into
   |   3D  $77$  tensor according to Seq.
3  end
4  Initialize parameters in BiLSTM-Attention
5   $e \leftarrow 0$ ;
6  while  $e < \text{Epoch}$  do
7  |   Learning  $h_t$  by independent and opposite LSTMs according to
   |   formulas (6)~(11)
8  |   Create the Stack Encoding Vector SEV $_t$  by computing the weighted
   |   sum.
9  |   A concatenation layer is applied to generate the attentional vector via
   |    $\tilde{h}_t = \tanh(W_t[\text{SEV}_t; h_t])$ 
10 |   Obtaining weight  $w'_t$  from attention layer through Softmax by
   |   formula (17).
11 |   Pass  $\tilde{h}_t$  and  $w'_t$  through a regular Dense layer, call Softmax() to
   |   obtain the predicted  $\hat{y}$ 
12 |    $e \leftarrow e + 1$ ;
13 end while
14 Export M for Trading Strategy

```

**3.3 Label design**

The estimated labels designed by this research include two types: (A) the short-term future trend and (B) the estimate of the stock price rise and fall next day, as follows:

(A) Short-term Future Trend:

This type of label is divided into three categories, including (1) the future trend is bullish, (2) the future trend is flat, and (3) the future trend is bearish, which are defined as follows:

$$\begin{cases} (1) : MA(n)_t^+ - MA(n)_t^- > MA(n)_t^- \times \alpha \\ (2) : |(MA(n)_t^- \times \alpha)| \geq |MA(n)_t^- - MA(n)_t^-| \\ (3) : MA(n)_t^+ - MA(n)_t^- < -MA(n)_t^- \times \alpha \end{cases} \tag{18}$$

where  $MA(n)_t^-$  denotes the moving average of stock prices in the past  $n$  days at day  $t$ ,  $MA(n)_t^+$  represents the moving average of stock prices in the next  $n$  days at day  $t$ , and  $\alpha$  is the threshold for judging whether to rise or fall. By default,  $n = 5$  and  $\alpha = 0.5\%$ .

(B) The next day price rises and falls:

This type of label compares the prices of the next day and the previous day within a certain percentage and is

defined as follows, where  $Close_t$  is the closing price at day  $t$ , and  $\alpha = 0.5\%$  by default:

$$\begin{cases} (1) : Close_{t+1} - Close_t > Close_t \times \alpha \\ (2) : |Close_t \times \alpha| \geq |Close_{t+1} - Close_t| \\ (3) : Close_{t+1} - Close_t < -Close_t \times \alpha \end{cases} \tag{19}$$

**3.4 Trading strategy design**

This research uses the output probability of Softmax to implement the trading strategy. The concept is as follows:

$$\text{Softmax}_{up} > x \tag{20}$$

$$\text{Softmax}_{down} > y \tag{21}$$

If the predicted increase probability of the output classification is greater than a threshold, it is a possible timing to buy. Similarly, if the predicted decline probability in the model is greater than a threshold, it is a possible timing to sell. Considering that real market transactions will include stop loss, profit stop and consumer cash flow, etc., to simplify the problem and focus on the effectiveness of the deep network model in the design of trading strategies, this study will analyze the following two trading strategies:

*Strategy\_A:* The market entry condition is fixed (while the probability  $\geq 35\%$ ), and the exit timing is set by the experiment.

*Strategy\_B:* The exit timing is fixed (while the probability  $\geq 35\%$ ), and the entry timing is set by the experiment.

**4 Experimental results and discussion**

**4.1 Experimental setting**

This study adopts Yuanta/P-shares Taiwan Top 50 ETF (TPE0050) as the research subject. TPE0050 is the earliest securities ETF issued in Taiwan and the largest ETF in the Taiwan stock market. The investment subject of TPE0050 is the top 50 large-cap stocks of the Taiwan Capitalization Weighted Stock Index.

The experiment data are collected from the Taiwan Stock Exchange (TWSE), and the transaction date used in this experiment is from 2014/01/04 to 2021/05/31, with a total of 1807 trading days. The data are divided into 60% training set, 20% for validation, and the last 20% for testing. The number of data is 1084, 361, and 362, respectively. In addition, the time interval for the backtest of the trading strategy is 2017/01/16 ~ 2019/10/14, a total of 677 trading days, which is a section where stock prices fluctuate more intensely.

**Table 1** Experimental combination of models, technical indicators and predicted labels, where AttBiLSTM denotes the proposed Attention-based BiLSTM model, STFT denotes the Short-term Future Trend, and NDP represents the Next Day Price prediction

Models	Model Features	Labels
Fundamental LSTM/AttBiLSTM	(I) Open + Close + Highest + Lowest + Volume (II) Open + Close + Highest + Lowest + Volume + Normalized KD (III) Open + Close + Highest + Lowest + Volume + Normalized RSI (IV) Open + Close + Highest + Lowest + Volume + Normalized BIAS (V) Open + Close + Highest + Lowest + Volume + Normalized W%R (VI) Open + Close + Highest + Lowest + Volume + Normalized MACD	STFT/NDP

**Table 2** Experimental results of STFT in fundamental LSTMs

	Trend	Exp. (I) (%)	Exp. (II) (%)	Exp. (III) (%)	Exp. (IV) (%)	Exp. (V) (%)	Exp. (VI) (%)
Precision	Up	29.63	51.89	31.84	49.78	34.09	32.87
	Flat	68.55	73.14	68.99	71.97	70.77	72.58
	Down	24.06	43.86	24.06	23.81	34.52	42.31
Recall	Up	11.62	21.74	12.89	19.82	14.71	14.61
	Flat	71.89	80.49	71.89	75.36	75.53	74.67
	Down	8.30	15.72	8.41	9.74	10.98	16.67
F1-Measure	Up	16.69	30.64	18.35	28.35	20.55	20.23
	Flat	70.18	76.64	70.41	73.62	73.07	73.61
	Down	12.35	23.15	12.47	13.82	16.67	23.91
Accuracy		57.40	66.80	57.87	62.60	61.33	62.67

The experimental hardware uses Asus Esc8000 G3 host, Xeon E5-2600 v3 processor, 4 GB memory, and NVIDIA GeForce GTX 1080ti × 8 GPUs with Ubuntu 18.04 LTS OS. The batch size is fixed at 32, epochs are between 50 and 200, and the parameter with the highest validation accuracy is selected as the prediction model. As shown in Table 1, the experimental settings are (a) fundamental LSTM and attention-based BiLSTM, (b) five different technical indicators with (c) two predicted labels cross-tests. Finally, a combination model with higher accuracy is selected for trading strategy experiments and performs the backtest to calculate the return on investment. The measures of this study are recall, precision, F1 measure, and accuracy, which are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{23}$$

$$F1 = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{24}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{25}$$

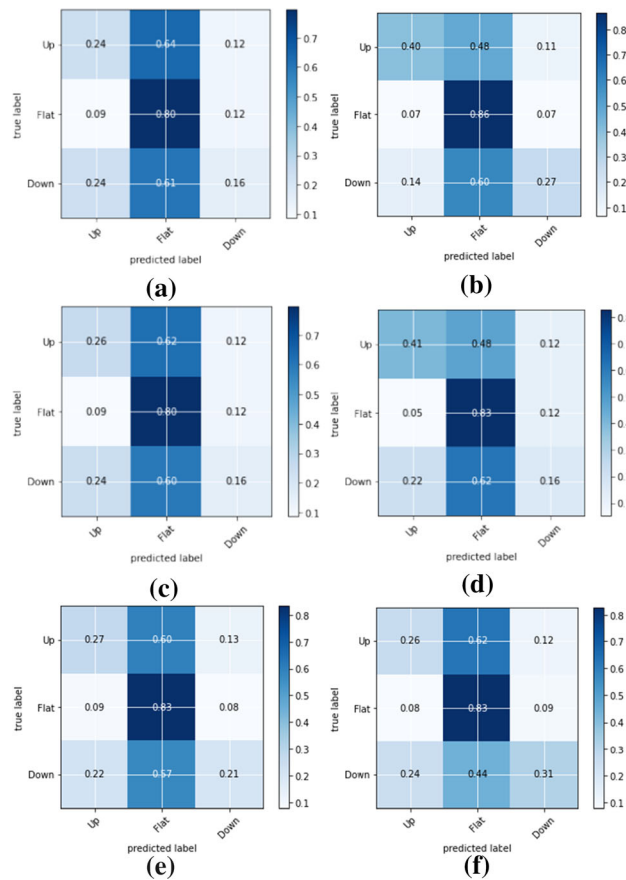
where TP, TN, FP, and FN represent the true positive, true negative, false positive, and false negative, respectively. F1 measure is the harmonic mean of recall and precision.

### 4.2 Experimental results of four-layer LSTM in labels STFT/NDP

The experimental results of fundamental LSTM for STFT are shown in Table 2 and Fig. 6. The accuracy of the experiment with is between 57.40 and 66.80%. The flat prediction results are generally higher in recall, precision, and F1 measure. The downward trend prediction is relatively low.

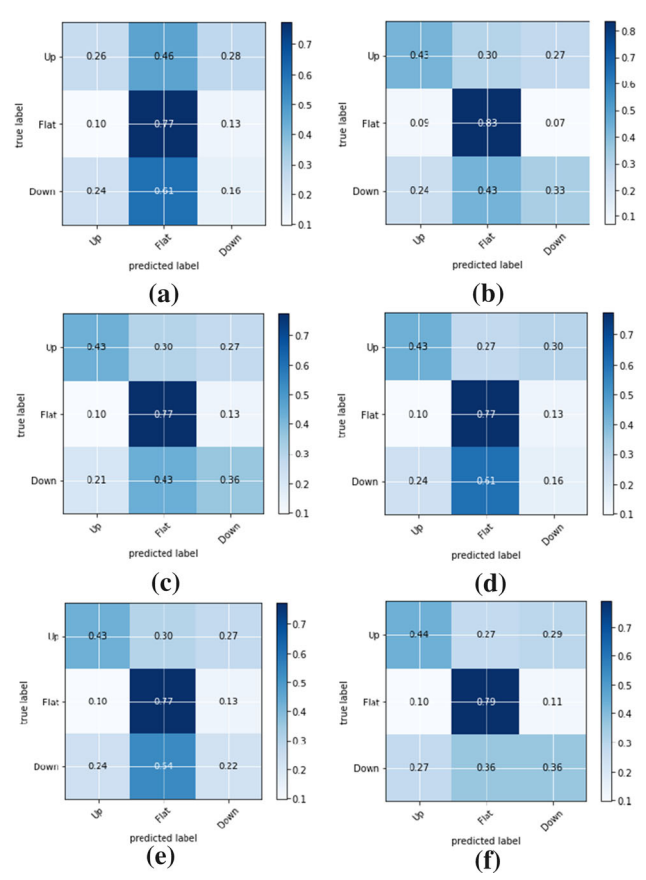
In these six experiments, the KD indicator performs better in forecasting, which achieved 66.80% prediction accuracy. Among various technical indicators, the prediction performance of RSI is relatively low.

The experimental results of fundamental LSTM for NDP are shown in Table 3 and Fig. 7. The experimental accuracy is between 52.73 and 63.80%. The result shows that in the NDP problem, KD index also obtains better performance. The overall accuracy is still greater than 60%.



**Fig. 6** Confusion Matrix of fundamental LSTM for STFT, (a) ~ (f) represent the results of features (I) to (VI), respectively

The experimental results of AttBiLSTM for STFT/NDP are shown in Tables 4, 5 and Figs. 8, 9. In STFT case, the experimental accuracy is between 62.00 and 68.83%, and the average accuracy is 64.70%, and 60.07–4.76% in NDP test. Compared with the fundamental LSTM model, there is



**Fig. 7** Confusion Matrix of fundamental LSTM for NDP, (a)–(f) represent the results of features (I) to (VI), respectively

a significant improvement in classification accuracy. The experimental results show that the KD indicator performs better than other types of technical indicators in both cases, followed by BIAS and MACD. In AttBiLSTM model, the improvements in average accuracy are 3.35 and 4.52 percentage points in STFT and NDP cases.

**Table 3** Experimental results of NDP in fundamental LSTMs

	Trend	Exp. (I) (%)	Exp. (II) (%)	Exp. (III) (%)	Exp. (IV) (%)	Exp. (V) (%)	Exp. (VI) (%)
Precision	Up	38.21	51.31	52.33	50.81	24.88	49.69
	Flat	65.40	75.00	73.51	70.33	64.36	76.44
	Down	17.51	37.40	33.23	17.05	13.49	34.09
Recall	Up	14.44	28.49	26.12	24.12	50.81	28.17
	Flat	64.17	69.60	64.94	63.73	71.10	64.56
	Down	8.30	19.83	21.58	9.53	23.72	22.25
F1-Measure	Up	20.96	36.64	34.85	32.71	33.40	35.96
	Flat	64.78	72.20	68.96	66.87	67.56	70.00
	Down	11.26	25.92	26.16	12.23	17.20	26.92
Accuracy		52.73	63.80	60.87	56.93	58.27	62.00



**Table 4** Experimental results of STFT in AttBiLSTM

	Trend	Exp. (I) (%)	Exp. (II) (%)	Exp. (III) (%)	Exp. (IV) (%)	Exp. (V) (%)	Exp. (VI) (%)
Precision	Up	41.94	46.91	41.94	46.27	48.48	41.23
	Flat	72.40	77.14	73.49	74.63	74.40	74.42
	Down	30.61	51.98	36.72	36.25	21.74	43.15
Recall	Up	20.35	25.28	20.76	24.90	25.67	19.54
	Flat	76.48	78.19	75.53	76.48	70.17	76.48
	Down	9.11	22.15	13.16	12.24	9.93	17.21
F1-Measure	Up	27.40	32.85	27.77	32.38	33.57	26.52
	Flat	74.38	77.66	74.49	75.54	72.22	75.44
	Down	14.04	31.07	19.37	18.30	13.64	24.60
Accuracy		63.27	68.83	63.93	65.47	62.00	65.27

**Table 5** Experimental results of NDP in AttBiLSTM

	Trend	Exp. (I) (%)	Exp. (II) (%)	Exp. (III) (%)	Exp. (IV) (%)	Exp. (V) (%)	Exp. (VI) (%)
Precision	Up	46.85	53.37	50.35	27.45	49.67	48.47
	Flat	72.99	78.44	73.80	66.14	73.20	77.01
	Down	34.20	39.43	36.28	23.86	35.58	41.92
Recall	Up	23.06	31.58	24.78	50.33	26.09	26.63
	Flat	65.16	66.18	65.81	75.76	67.89	65.94
	Down	20.92	27.06	23.37	37.46	19.15	28.46
F1-Measure	Up	30.91	39.68	33.22	35.52	34.21	34.38
	Flat	68.85	71.79	69.57	70.63	70.45	71.05
	Down	25.96	32.09	28.43	29.15	24.90	33.90
Accuracy		60.07	64.76	61.40	62.73	61.80	63.58

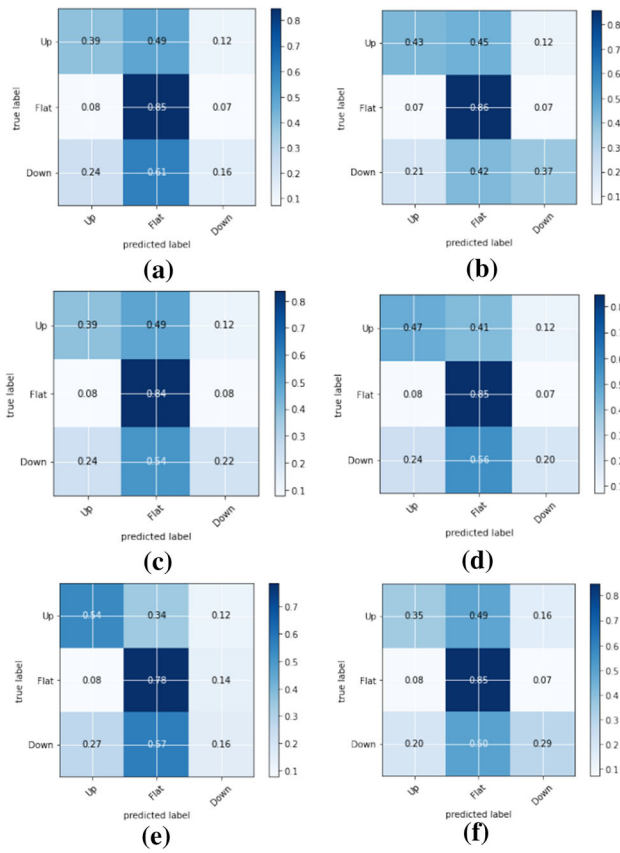
### 4.3 Experiments on trading strategy and return on investment (ROI)

This section discusses the return on investment (ROI) for different entry and exit conditions. According to the trading strategy settings described in Sect. 3, this study will backtest the proposed model on TPE0050 and evaluate the corresponding ROI. The backtest condition is a linear backtest with a single buy and sell. The backtest date is 2017/01/16–2019/10/14, which is the range of large price fluctuations. The experimental results of Sect. 4.2 show that the AttBiLSTM model has a better predictive effect on STFT problem. In this section, the combination of AttBiLSTM + STFT is used to analyze the trading strategy. The ROI in this study is set as follows:

$$\text{ROI} = \frac{\text{Selling price} - \text{Buying price}}{\text{Buying price}} \quad (26)$$

#### 4.3.1 Strategy\_A & Strategy\_B

In the proposed trading strategy design, the approach of *Strategy\_A* is to enter the market when the predicted *price\_up* probability  $\geq 35\%$ . Strategy A fixes the buying timing to the rising probability of 0.35, while the selling probability is divided into 0.2, 0.25, 0.3, and 0.35 for comparison (as shown in Table 6 and Fig. 10), the results show that the probability of 0.3–0.35 has a higher ROI, and AttBiLSTM + KD indicator obtains the highest ROI of 42.57%. *Strategy\_B* has similar experimental result with *Strategy\_A*, the results are shown in Table 7 and Fig. 11. When the falling probability is fixed at 35% and the rising probability is 35%, the model with KD indicator reaches the highest ROI of 42.74%. Figure 12 shows the backtest of buying/selling simulation from 2018/10/31 to 2019/05/16.

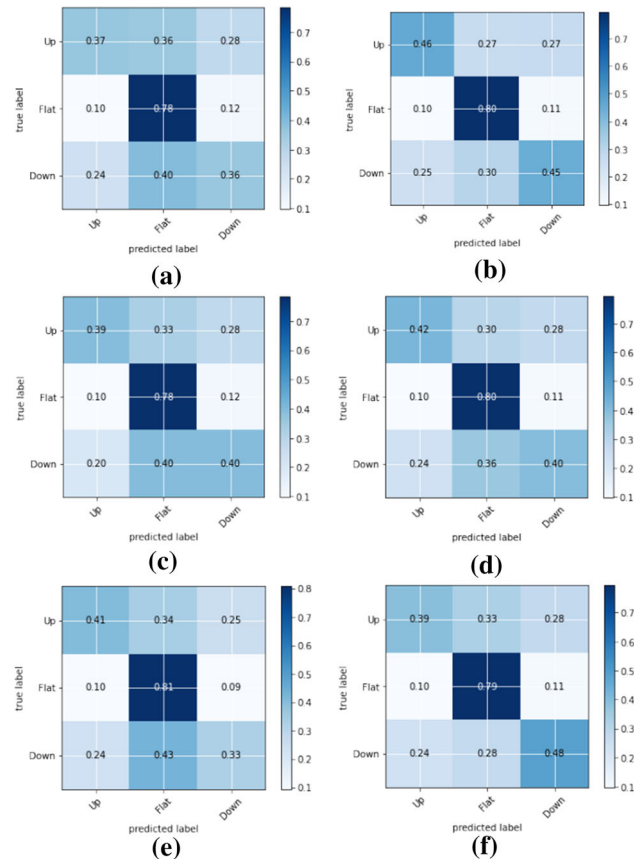


**Fig. 8** Confusion Matrix of fundamental AttBiLSTM for STFT, (a)–(f) represent the results of features (I) to (VI), respectively

### 5 Conclusions and future work

This research explores the effectiveness of attention-based time series neural networks applied to stock forecasting and financial commodity trading strategy design. This research proposes the concept of attention-based BiLSTM combined with trading strategy design and verifies the effectiveness of different technical indicators in the model. This study uses stochastic oscillator (KD indicator), RSI, BIAS, W%R, and MACD, which are commonly used in stock technical analysis, to perform the short-term trend and next-day rise/fall predictions. This research concluded the following conclusions:

1. Taking TPE:0050 as the research target, the Att-BiLSTM model performs better than the fundamental LSTM model.
2. Among the several technical indicators based on the moving average concept, the KD indicator has a better overall effect, including classification forecasting and measurement of return on investment.
3. The results of the backtest experiment show that the proposed deep trading strategy can achieve a maximum ROI of 42.74%



**Fig. 9** Confusion Matrix of fundamental AttBiLSTM for NDP, (a)–(f) represent the results of features (I) to (VI), respectively

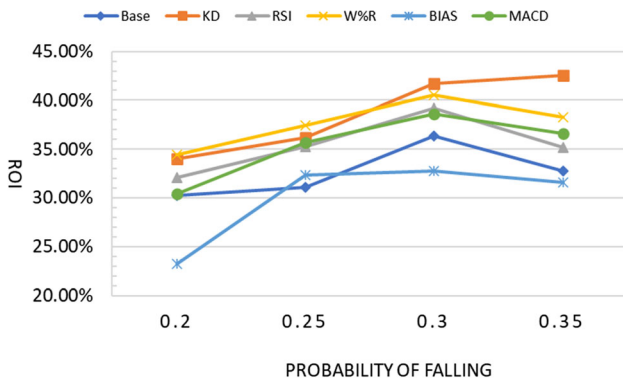
**Table 6** Strategy\_A, ROI of different probabilities of falling

TAs	Falling Probability				Average
	0.2 (%)	0.25 (%)	0.3 (%)	0.35 (%)	
Base	30.29	31.06	36.33	32.79	32.62
KD	34.03	36.21	41.72	42.57	38.63
RSI	32.10	35.29	39.18	35.20	35.44
WR	34.44	37.42	40.50	38.27	37.66
BIAS	23.23	32.31	32.78	31.56	29.97
MACD	30.43	35.65	38.61	36.62	35.33
Max	34.44	37.42	41.72	42.57	–

4. The probability output of the deep network model can be directly applied to the trading strategy of financial commodities.

Future works have the following extensions:

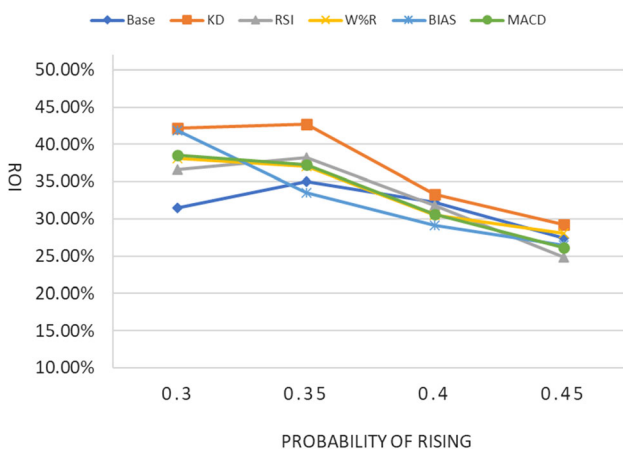
1. The technical indicators used in this study are mainly moving averages. In the future, more different types of technical indicator analysis can be included.



**Fig. 10** The change trend of ROI in different falling probability with fixed *price\_up* probability > = 35%

**Table 7** Strategy\_B, ROI of different probabilities of rising

TAs	Falling Probability				Average
	0.3 (%)	0.35 (%)	0.4 (%)	0.45 (%)	
Base	31.43	35.02	32.23	27.37	31.51
KD	42.14	42.74	33.34	29.25	36.87
RSI	36.57	38.23	31.78	24.83	32.85
WR	38.14	37.01	30.47	28.02	33.41
BIAS	41.82	33.56	29.08	26.43	32.73
MACD	38.50	37.25	30.59	26.11	33.11
Max	42.14	42.74	33.34	29.25	—



**Fig. 11** The change trend of ROI in different rising probability with fixed *price\_down* probability > = 35%

2. More time series or non-time series networks can be considered for trading strategy design, such as CNN, attention-based CNN, GAN, and transformer.
3. In the future, the concepts proposed in this study can be used in advanced financial commodity transaction types, such as stretchable financial commodities, day-trading, futures trading, and option transaction.



**Fig. 12** Backtest simulation of buying/selling from 2018/10/31 to 2019/05/16

**Acknowledgements** The author would like to thank the Ministry of Science and Technology (MOST) under MOST 110-2221-E-130-008 for their support during this study.

**Declarations**

**Conflict of interest** No conflict of interest exists in the submission of this manuscript, and manuscript is approved by all authors for publication.

**References**

1. Cootner PH (ed) (1967) The random character of stock market prices. MIT Press, Cambridge (MA)
2. Brealey RA, Myers SC, Allen F, Mohanty P (2012) Principles of corporate finance. Tata McGraw-Hill Education, New York
3. Abarbanell JS, Bushee BJ (1997) Fundamental analysis, future earnings, and stock prices. J Account Res 35(1):1–24
4. De Long JB, Shleifer A, Summers LH, Waldmann RJ (1990) Positive feedback investment strategies and destabilizing rational speculation. J Finance 45(2):379–395
5. Murphy JJ (1999) Technical analysis of the financial markets: a comprehensive guide to trading methods and applications. Penguin, New York
6. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553), 436–444.
7. Cao C, Liu F, Tan H, Song D, Shu W, Li W, Zhou X, Bo Y, Xie Z (2018) Deep learning and its applications in biomedicine. Genom Proteomics Bioinform 16(1):17–32
8. Chiu PS, Chang JW, Lee MC, Chen CH, Lee DS (2020) Enabling intelligent environment by the design of emotionally aware virtual assistant: a case of smart campus. IEEE Access 8:62032–62041
9. Lee MC, Chiu SY, Chang JW (2017) A deep convolutional neural network based Chinese menu recognition app. Inf Process Lett 128:14–20
10. Lee MC, Chiang SY, Yeh SC, Wen TF (2020) Enabling intelligent environment recognition and companion Chatbot using deep neural network. Multimedia Tools and Appl 79(27):19629–19657
11. Miiikkulainen R, Liang J, Meyerson E, Rawal A, Fink D, Francon O, Raju B, Shahrzad H, Navruzyan A, Duffy N, Hodjat B (2019) Evolving deep neural networks. In: Artificial Intelligence in the Age of Neural Networks and Brain Computing. Academic Press, New York, pp 293–312.
12. Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX (2019) Deep learning and its applications to machine health monitoring. Mech Syst Signal Process 115:213–237

13. Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S (2010). Recurrent neural network based language model. In *Interspeech* (Vol. 2, p. 3).
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
15. Choi E, Bahadori MT, Sun J, Kulas J, Schuetz A, Stewart W (2016) Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In: *Advances in Neural Information Processing Systems*, pp 3504–3512.
16. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: *Advances in neural information processing systems*, pp 5998–6008.
17. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. *Data Min Knowl Disc* 33(4):917–963
18. Gamboa JCB (2017) Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*.
19. Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. *Appl Soft Comput* 90:106181.
20. Qi Y, Xiao J (2018) Fintech: AI powers financial services to improve people’s lives. *Commun ACM* 61(11):65–69
21. Nelson DM, Pereira AC, de Oliveira RA (2017) Stock market’s price movement prediction with LSTM neural networks. In: *2017 International joint conference on neural networks (IJCNN)*. IEEE, New York, pp 1419–1426.
22. Chen W, Yeo CK, Lau CT, Lee BS (2018) Leveraging social media news to predict stock index movement using RNN-boost. *Data Knowl Eng* 118:14–24
23. Cakra YE, Trisedya BD (2015) Stock price prediction using linear regression based on sentiment analysis. In *2015 international conference on advanced computer science and information systems (ICACSIS)*. IEEE, New York, pp. 147–154.
24. Li H, Shen Y, Zhu Y (2018) Stock price prediction using attention-based multi-input LSTM. In: *Asian Conference on Machine Learning*. PMLR, pp 454–469
25. Li C, Song D, Tao D (2019) Multi-task recurrent neural networks and higher-order Markov random fields for stock price movement prediction: Multi-task RNN and higher-order MRFs for stock price classification. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1141–1151.
26. Livieris IE, Pintelas E, Pintelas P (2020) A CNN–LSTM model for gold price time-series forecasting. *Neural Comput Appl* 32(23):17351–17360
27. Hu Z (2021) Crude oil price prediction using CEEMDAN and LSTM-attention with news sentiment index. *Oil Gas Sci Technol Revue d’IFP Energies Nouvelles* 76:28
28. Lee MC, Chang JW, Hung JC, Chen BL (2021) Exploring the effectiveness of deep neural networks with technical analysis applied to stock market prediction. *Comput Sci Inf Syst* 18(2):401–418
29. Russell S, Norvig P (2002) *Artificial intelligence: a modern approach*. \*\*
30. Bishop CM (2006) *Pattern recognition*. *Machine Learn* 128(9).
31. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert A, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, van den Driessche G, Graepel T, Hassabis D (2017) Mastering the game of go without human knowledge. *Nature* 50(7676):354–359.
32. ACM (2018) *Fathers of the Deep Learning Revolution Receive ACM A.M. Turing Award—Bengio, Hinton and LeCun Ushered in Major Breakthroughs in Artificial Intelligence*. <https://awards.acm.org/about/2018-turing>
33. Stilgoe J (2018) Machine learning, social learning and the governance of self-driving cars. *Soc Stud Sci* 48(1):25–56
34. Carlini N, Wagner D (2018) Audio adversarial examples: Targeted attacks on speech-to-text. In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, New York, pp. 1–7.
35. Xiong W, Wu L, Allewa F, Droppo J, Huang X, Stolcke A (2018) The Microsoft 2017 conversational speech recognition system. In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, New York, pp. 5934–5938
36. Amanullah MA, Habeeb RAA, Nasaruddin FH, Gani A, Ahmed E, Nainar ASM, Akim NM, Imran M (2020) Deep learning and big data technologies for IoT security. *Comput Commun* 151:495–517
37. Haddad Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. In: *International conference on machine learning*, pp. 2048–2057.
38. Deng J, Guo J, Xue N, Zafeiriou S (2019) Arcface: Additive angular margin loss for deep face recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 4690–4699.
39. Habibzadeh H, Boggio-Dandry A, Qin Z, Soyata T, Kantarci B, Mouftah HT (2018) Soft sensing in smart cities: Handling 3Vs using recommender systems, machine intelligence, and data analytics. *IEEE Commun Mag* 56(2):78–86
40. Lee S, Choi DH (2020) Energy management of smart home with home appliances, energy storage system and electric vehicle: a hierarchical deep reinforcement learning approach. *Sensors* 20(7):2157
41. Mohammadi M, Al-Fuqaha A (2018) Enabling cognitive smart cities using big data and machine learning: approaches and challenges. *IEEE Commun Mag* 56(2):94–101
42. Shi Q, Zhang Z, He T, Sun Z, Wang B, Feng Y, Shan X, Salam B, Lee C (2020) Deep learning enabled smart mats as a scalable floor monitoring system. *Nat Commun* 11(1):1–11
43. Tian Z, Cui Y, An L, Su S, Yin X, Yin L, Cui X (2018) A real-time correlation of host-level events in cyber range service for smart campus. *IEEE Access* 6:35355–35364
44. Peng J, Zhou Y, Sun X, Su J, Ji R (2018) Social media based topic modeling for smart campus: a deep topical correlation analysis method. *IEEE Access* 7:7555–7564
45. Cheng Y (2019) *Semi-supervised learning for neural machine translation*. In: *Joint Training for Neural Machine Translation*. Springer, Singapore, pp. 25–40.
46. Liu Y, Gu J, Goyal N, Li X, Edunov S, Ghazvininejad M, Lewis M, Zettlemoyer L (2020) Multilingual denoising pre-training for neural machine translation. *Trans Assoc Comput Linguistics* 8:726–742
47. Schubert K (2019) *Metataxis: Contrastive dependency syntax for machine translation*, vol 2. Walter de Gruyter GmbH & Co KG.
48. Öztürk Ş (2021) Convolutional neural network based dictionary learning to create hash codes for content-based image retrieval. *Proc Comput Sci* 183:624–629
49. Özkaya U, Öztürk Ş, Melgani F, Seyfi L (2021) Residual CNN+Bi-LSTM model to analyze GPR B scan images. *Automation Const* 123:103525.
50. Öztürk Ş (2021) Class-driven content-based medical image retrieval using hash codes of deep features. *Biomed Signal Process Control* 68:102601.
51. Alshemali B, Kalita J (2020) Improving the reliability of deep neural networks in NLP: a review. *knowledge-based systems*, p 191, 105210.
52. Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag* 13(3):55–75

53. Ng A, Ngiam J, Foo CY, Mai Y, Suen C (2012) UFLDL tutorial. Chapters available at [http://deeplearningstanford.edu/wiki/index.php/UFLDL\\_Tutorial](http://deeplearningstanford.edu/wiki/index.php/UFLDL_Tutorial).
54. Werbos P (1974) Beyond regression: “ new tools for prediction and analysis in the behavioral sciences. PhD dissertation, Harvard University.
55. Widrow B, Hoff ME (1960) Adaptive switching circuits. Stanford Univ Ca Stanford Electronics Labs.
56. Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S (2016) Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 961–971.
57. Skrobek D, Krzywanski J, Sosnowski M, Kulakowska A, Zylka A, Grabowska K, Ciesielska K, Nowak W (2020) Prediction of sorption processes using the deep learning methods (Long Short-Term Memory). *Energies* 13(24):6601
58. Chimmula VKR, Zhang L (2020) Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons Fractals*, 135:109864.
59. Mnih V, Heess N, Graves A (2014) Recurrent models of visual attention. In: Advances in neural information processing systems, pp 2204–2212.
60. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473).
61. Sutskever I, Vinyals O, Le Q.V (2014) Sequence to sequence learning with neural networks. arXiv preprint [arXiv:1409.3215](https://arxiv.org/abs/1409.3215).
62. Wilder JW (1978) New concepts in technical trading systems. *Trend Res.*
63. Abdulali A (2006) *The Bias Ratio™: Measuring the Shape of Fraud*. Protégé Partners, New York
64. Williams LR (1979) *How i made one million dollars last year*. Windsor Books, Trading Commodities
65. Appel G (2005) *Technical analysis: power tools for active investors*. FT Press, Upper Saddle River.

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