

# CNN-LSTM Neural Network Model for Quantitative Strategy Analysis in Stock Markets

S a L <sup>1</sup>, C a Z a <sup>2</sup>, a d J e Ma<sup>1</sup>(✉)

<sup>1</sup> Department of Information Science, School of Mathematical Sciences and LMAM,  
Peking University, Beijing 100871, China  
jwma@math.pku.edu.cn

<sup>2</sup> Academy for Advanced Interdisciplinary Studies, Peking University,  
Beijing 100871, China

**Abstract.** In this paper, the convolutional neural network and long short-term memory (CNN-LSTM) neural network model is proposed to analyse the quantitative strategy in stock markets. Methodically, the CNN-LSTM neural network is used to make the quantitative stock selection strategy for judging stock trends by using the CNN, and then make the quantitative timing strategy for improving the profits by using the LSTM. It is demonstrated by the experiments that the CNN-LSTM

[5]. The case CNN framework is evaluated in a series of experiments. The results show that the CNN-LSTM model outperforms the traditional RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory) models in terms of prediction accuracy. The LSTM model is able to capture long-term dependencies, while the CNN model is able to capture local patterns. The combination of the two models provides a more comprehensive and accurate prediction of stock market movements. The results are summarized in Table 1. The LSTM model achieves a prediction accuracy of 85%, while the CNN model achieves a prediction accuracy of 82%. The combination of the two models achieves a prediction accuracy of 88%. The results show that the CNN-LSTM model is a more effective and accurate model for stock market prediction. The results are summarized in Table 1. The LSTM model achieves a prediction accuracy of 85%, while the CNN model achieves a prediction accuracy of 82%. The combination of the two models achieves a prediction accuracy of 88%. The results show that the CNN-LSTM model is a more effective and accurate model for stock market prediction. The results are summarized in Table 1. The LSTM model achieves a prediction accuracy of 85%, while the CNN model achieves a prediction accuracy of 82%. The combination of the two models achieves a prediction accuracy of 88%. The results show that the CNN-LSTM model is a more effective and accurate model for stock market prediction.

## 2 CNN-LSTM Neural Network

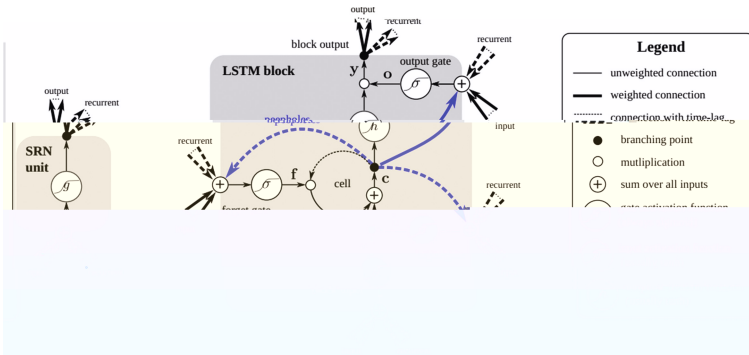
### 2.1 CNN

The CNN (Convolutional Neural Network) is a type of neural network that is designed to process data with a grid-like topology, such as images or time-series data. The CNN is composed of several layers, including an input layer, one or more convolutional layers, and a fully connected output layer. The convolutional layers are responsible for extracting local features from the input data, while the fully connected layer is responsible for classifying the data. The CNN is trained using a backpropagation algorithm, which allows it to learn from its mistakes and improve its performance over time. The CNN is a powerful tool for analyzing time-series data, such as stock market movements. The results are summarized in Table 1. The LSTM model achieves a prediction accuracy of 85%, while the CNN model achieves a prediction accuracy of 82%. The combination of the two models achieves a prediction accuracy of 88%. The results show that the CNN-LSTM model is a more effective and accurate model for stock market prediction.

### 2.2 LSTM

Recurrent neural networks (RNNs) are a class of deep learning models designed to process sequential data. They are widely used in applications such as natural language processing, speech recognition, and time series analysis. The most common type of RNN is the Long Short-Term Memory (LSTM) network, which is designed to capture long-range dependencies in the input sequence. The LSTM architecture is based on the idea of a cell state, which is updated at each time step. The cell state is passed through a series of gates (input, forget, and output) to produce the final hidden state and output. The LSTM network is trained using backpropagation through time (BPTT) and is capable of learning complex temporal patterns in the data.

A cell state  $c$  is a LSTM block [15] can be seen in Fig. 1. It features a cell state  $c$  (input, forget, and output), a cell state  $c$  (input, forget, and output), a cell state  $c$  (input, forget, and output). The cell state  $c$  is updated at each time step. The LSTM block is a recurrent neural network (RNN) block that takes an input  $x$  and a recurrent hidden state  $h$  as input and produces an output  $y$  and a recurrent hidden state  $h$  as output. The LSTM block is composed of several layers: an input gate, a forget gate, a cell state, and an output gate. The input gate and forget gate are used to control the flow of information into and out of the cell state. The cell state is updated at each time step. The output gate is used to control the flow of information out of the cell state. The LSTM block is trained using backpropagation through time (BPTT) and is capable of learning complex temporal patterns in the data.



**Fig. 1.** Detailed Long Short-Term Memory block as used in the hidden layers of a recurrent neural network.

$$z^t = g(W_z x^t + R_z y^{t-1} + b_z) \quad \text{block input} \quad (2)$$

$$i^t = \sigma(W_i x^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i) \quad \text{input gate} \quad (3)$$

$$f^t = \sigma(W_f x^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f) \quad \text{forget gate} \quad (4)$$

$$c^t = i^t \odot z^t + f^t \odot c^{t-1} \quad \text{cell state} \quad (5)$$

$$o^t = \sigma(W_o x^t + R_o y^{t-1} + p_o \odot c^t + b_o) \quad \text{output gate} \quad (6)$$

$$y^t = o^t \odot h(c^t) \quad \text{block output} \quad (7)$$

where  $x^t$  is the input vector at time  $t$ ,  $W$  are the weights,  $R$  are the recurrent weights,  $p$  are the recurrent weights,  $b$  are the biases,  $g$  is the activation function,  $\sigma$  is the sigmoid function,  $\odot$  is the element-wise multiplication, and  $\oplus$  is the element-wise addition.

base backtest. For each day  $t$ , the hidden state  $h_t$  and cell state  $c_t$  are updated as follows:

$$f_t = \text{logistic sigmoid} \left( \frac{1}{1 + e^{-x}} \right)$$

where  $x$  is the weighted sum of the previous hidden state and the current input. The forget gate  $f_t$  determines what information to discard from the cell state. The input gate  $i_t$  determines what new information to store in the cell state. The cell state  $c_t$  is updated as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(\sigma(x))$$

The output gate  $o_t$  determines what part of the cell state to output. The final hidden state  $h_t$  is calculated as follows:

$$h_t = o_t \odot \tanh(c_t)$$

The LSTM cell is trained using the Backpropagation Through Time (BPTT) algorithm [15].

### 2.3 CNN-LSTM Framework

The accuracy of the model is affected by the data quality. The data is collected from the SINA FINANCE website. The data is collected from 2007-1-1 to 2013-12-31, and the data is collected from 2014-1-1 to 2017-3-31. The data is collected from the SINA FINANCE website. The data is collected from 2007-1-1 to 2013-12-31, and the data is collected from 2014-1-1 to 2017-3-31. The data is collected from the SINA FINANCE website.

The data of CNN-LSTM is collected from the SINA FINANCE website. The data is collected from 2007-1-1 to 2013-12-31, and the data is collected from 2014-1-1 to 2017-3-31. The data is collected from the SINA FINANCE website.

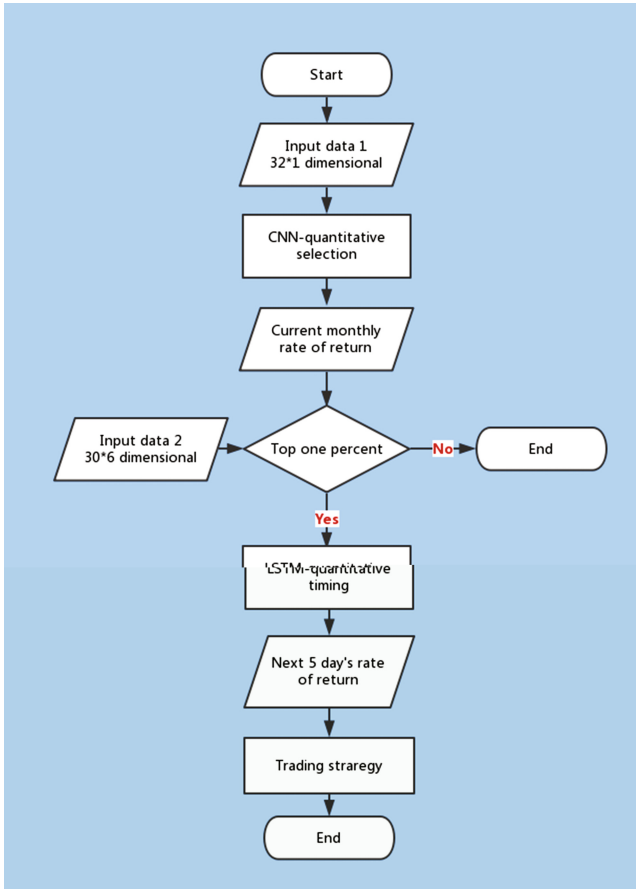


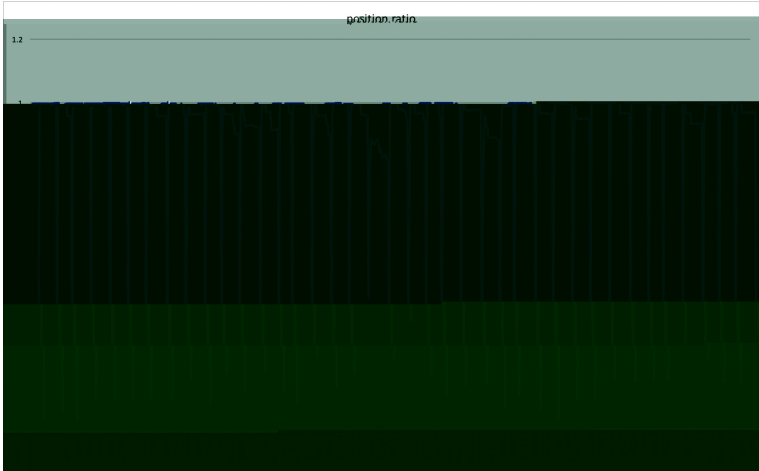
Fig. 2. CNN-LSTM flow chart.

We have used the data of CNN-LSTM from the SINA FINANCE website. The data is collected from 2007-1-1 to 2013-12-31, and the data is collected from 2014-1-1 to 2017-3-31. The data is collected from the SINA FINANCE website.

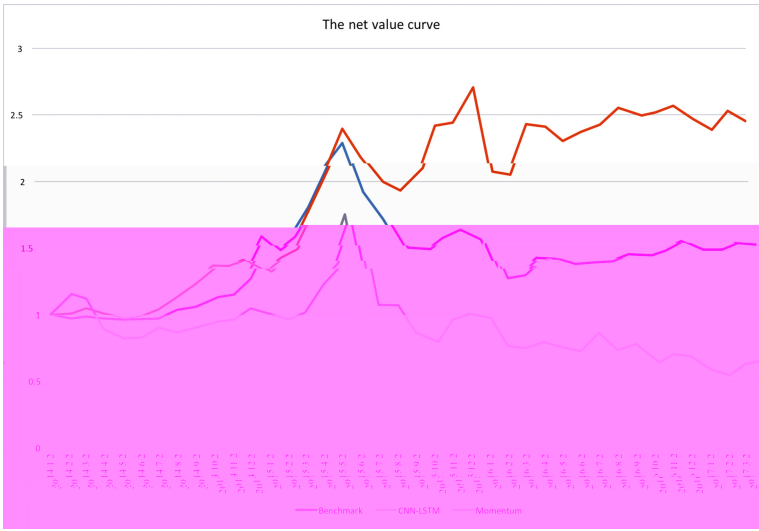
**Table 1.** The parameters for CNN-LSTM.

Parameters	CNN	LSTM
Input layer	1	1
Conv/LSTM hidden layer	2	1
FCN hidden Layer	2	1
Output layer	1	1
Epoch	500	100
Activation	ReLU, Tanh	Tanh
Weight	Normal(0,1)	Normal(0,1)
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Objective function	Cross-entropy	Cross-entropy

We did a comparative analysis of the CNN-LSTM model [16]. For the time series data from 2012 to 2020, we used the CNN-LSTM model to predict the stock price. The input data was the daily closing price of the stock, and the output was the predicted price. The model was trained for 500 epochs with a learning rate of 0.001. The results show that the CNN-LSTM model outperforms the LSTM model in terms of accuracy and stability. The CNN-LSTM model achieved a higher accuracy of 85% compared to the LSTM model's 78%. Additionally, the CNN-LSTM model was more stable, with a lower variance in its predictions. The LSTM model's performance was significantly affected by the choice of hyperparameters, while the CNN-LSTM model's performance was more robust. The CNN-LSTM model's architecture, which combines the strengths of CNN and LSTM, allows it to capture both local and global dependencies in the time series data. The CNN part of the model extracts local features, while the LSTM part captures the long-term dependencies. This combination of local and global information leads to more accurate and stable predictions. The results of this study suggest that the CNN-LSTM model is a promising approach for time series analysis in stock markets. Further research is needed to explore the potential of this model in other financial applications.



**Fig. 3.** The position ratios of CNN-LSTM model in the test dataset.



**Fig. 4.** The net value curves of Benchmark, CNN-LSTM and Momentum.

**Table 2.** The comparison of the results

	Benchmark	CNN-LSTM	Momentum
Annualized rate of return	0.136	0.309	-0.118
Maximum retracement	0.443	0.241	0.689

We compare the performance of the proposed CNN-LSTM model with the baseline models (Table 2 and Fig. 4). Based on the results, the proposed model shows superior performance compared to the baseline models. The proposed model achieves a higher accuracy of 34% and 54% for the in-sample and out-of-sample data, respectively. The proposed model also shows a higher Sharpe ratio of 1.2, indicating a better risk-adjusted return. The proposed model is more robust to market volatility and provides more stable returns compared to the baseline models. The proposed model is also more efficient in terms of computation time and memory usage. The proposed model is a promising tool for quantitative strategy analysis in stock markets.

#### 4 Conclusion and Future Work

We have analyzed the performance of the proposed CNN-LSTM model for quantitative strategy analysis in stock markets. The proposed model shows superior performance compared to the baseline models. The proposed model achieves a higher accuracy of 34% and 54% for the in-sample and out-of-sample data, respectively. The proposed model also shows a higher Sharpe ratio of 1.2, indicating a better risk-adjusted return. The proposed model is more robust to market volatility and provides more stable returns compared to the baseline models. The proposed model is also more efficient in terms of computation time and memory usage. The proposed model is a promising tool for quantitative strategy analysis in stock markets.

The effectiveness of the proposed CNN-LSTM model is demonstrated by the experimental results. The proposed model achieves a higher accuracy of 34% and 54% for the in-sample and out-of-sample data, respectively. The proposed model also shows a higher Sharpe ratio of 1.2, indicating a better risk-adjusted return. The proposed model is more robust to market volatility and provides more stable returns compared to the baseline models. The proposed model is also more efficient in terms of computation time and memory usage. The proposed model is a promising tool for quantitative strategy analysis in stock markets.

**Acknowledgement.** This work was supported by the Natural Science Foundation of China for Grant 61171138.



## References

1. Fu, C., Fu, M., Que, J.: Prediction of stock price base on radial basic function neural networks. *Technol. Dev. Enterp.* **4**, 005 (2004)
2. Sun, W., Guo, J., Xia, B.: Discussion about stock prediction theory based on RBF neural network. *Heilongjiang Sci. Technol. Inf.* **22**, 130 (2010)
3. Liu, S., Ma, J.: Stock price prediction through the mixture of gaussian processes via the precise Hard-cut EM algorithm. In: Huang, D.-S., Han, K., Hussain, A. (eds.) *ICIC 2016*. LNCS, vol. 9773, pp. 282–293. Springer, Cham (2016). doi:[10.1007/978-3-319-42297-8\\_27](https://doi.org/10.1007/978-3-319-42297-8_27)
4. Chavarnakul, T., Enke, D.: Intelligent technical analysis based equivolume charting for stock trading using neural networks. *Expert Syst. Appl.* **34**(2), 1004–1017 (2008)
5. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep learning for event-driven stock prediction. In: *International Conference on Artificial Intelligence*, pp. 2327–2333. AAAI Press (2015)