Journal of computer science and software applications
https://www.mfacademia.org/
ISSN:2377-0430
Vol. 4, No. 3, 2024

Predicting Stock Market Trends Using LSTM Networks: Overcoming RNN Limitations for Improved Financial Forecasting

Jiajing Wang[1], Siyi Hong[2], Yuxin Dong[3] , Zichao Li[4], Jinxin Hu[5]
[1]Columbia University ,New York,USA
[2]Columbia University ,New York,USA
[3]Wake Forest University , Winston-Salemo,USA
[4]Canoakbit Alliance Inc, Oakville, Canada
[5]Arizona State University,Tempe, USA

Correspondence should be addressed to Yuxin Dong; dongyuxin94616@gmail.com

Abstract:
In recent years, stocks have increasingly attracted our attention. The inherent volatility of stock prices, often caused by national and social policies, makes it challenging for investors to achieve profitable returns in the stock market. With the rapid advancement of artificial intelligence, computers have become adept at handling complex mathematical problems. Consequently, efforts have been made to leverage computers' remarkable computational capabilities to analyze and predict stock market trends. A growing number of professionals are delving into technologies related to deep learning. Two prominent applications in this field are data classification and regression. Recurrent Neural Networks (RNNs) have demonstrated superior performance in processing sequential data compared to other neural networks. However, RNNs can face issues such as gradient explosion or gradient vanishing when handling large datasets. These problems can cause RNNs to forget earlier data, resulting in inaccurate predictions. To address these issues, the Long Short-Term Memory (LSTM) model, an enhanced version of RNN, was introduced. LSTM incorporates input gates, forget gates, and output gates, effectively mitigating data forgetting and gradient explosion problems. By harnessing the computational power of computers, it becomes possible to make informed predictions about stock movements.

Keywords:
Deep Learning; Stock Forecasting; Recurrent Neural Networks; LSTM Networks; Neural Networks.

1. Introduction
With the advancement of the economy, people are encountering more novel things and beginning to invest in financial markets, with stocks being a primary entry point. In the 21st century, stocks have become a prevalent method of financial investment, and making a profit in the stock market has become a common goal for investors. To achieve this, understanding stock trends is essential, which is why predicting stock market movements has attracted considerable attention from both society and academia. The volatility of stock trends, influenced by market fluctuations and significant changes due to policy shifts, makes it increasingly difficult for investors to predict stock movements and profit from their investments[1].
With the rapid advancement of artificial intelligence, an increasing number of experts and scholars are exploring technologies in the field of deep learning. Deep learning is widely applied in image recognition[2-3], image classification[4-7], object detection[8], natural language processing[9-10], speech recognition, text classification, and various other domains. Among these, neural networks can significantly enhance the accuracy of stock trend predictions [11]. It has been discovered that Recurrent Neural Networks (RNNs) excel in learning and processing sequential data compared to other neural networks. However, RNNs have a tendency to forget previous data too quickly when processing large

datasets [12]. To mitigate this issue, Long Short-Term Memory (LSTM) networks were introduced. LSTM networks incorporate input gates, output gates, and forget gates, effectively addressing the problems of data forgetting and gradient explosion encountered by RNNs [13]. By training RNNs with several years of stock trend data, we enable computers to analyze past stock trends and predict future movements over the next few days.

2. Related work

2.1. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) consist of an input layer, multiple hidden layers, and an output layer. The training data is initially divided into several equally sized arrays arranged in sequential order[14]. By sequentially feeding these arrays into the neural network for training, a network model is constructed. The training process is illustrated in Figure 1:connected, they constitute an artificial neural network, as exemplified by the diagram in Figure 1:

$X(t)$ denotes the t-th array in a set of divided arrays. When this array is fed into the Recurrent Neural Network (RNN) for training, the hidden layer constructs a description of the array, represented as $S(t)$. Upon inputting the $(t+1)$-th array $(t+1)$-th array, it adjusts its description based on $S(t)$. This connection between consecutive array inputs ensures unique accuracy for sequential data.

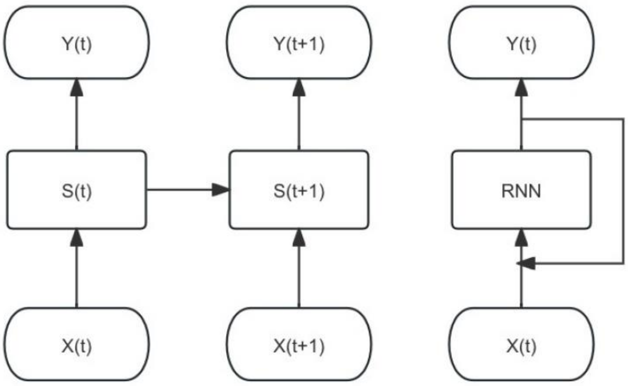

Figure1.The Training Process of Recurrent Neural Networks

However, RNNs often perform inadequately when processing large datasets [15]. They forget data rapidly, mainly due to gradient vanishing or gradient explosion caused by the significant difference between the results of earlier and later data during long sequence training, as shown in Figure 2.

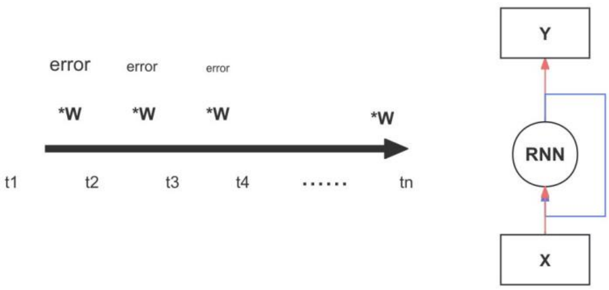

Figure2.Issues with Recurrent Neural Networks in Processing Long Sequences

When training on long data sequences, the initial result produced by the RNN is T, while the final result is R. The significant difference between R and T leads to errors. If the error is between 0 and 1, it diminishes to nearly zero as it is propagated through multiple layers of weights, resulting in gradient vanishing[16]. Conversely, if the error is greater than 1, it amplifies to near infinity through the layers, leading to gradient explosion. These issues can cause the RNN to make incorrect predictions. To mitigate these errors, Long Short-Term Memory (LSTM) networks are employed.

2.2. LSTM Recurrent Neural Networks

LSTM Recurrent Neural Networks (RNNs) are adept at handling long data sequences. The core concept of LSTM is that each neuron maintains the information stored in a cell, and three gates—input gate, output gate, and forget gate—are used to regulate the addition and removal of information from a cell[17]. This process is illustrated in Figure 3:

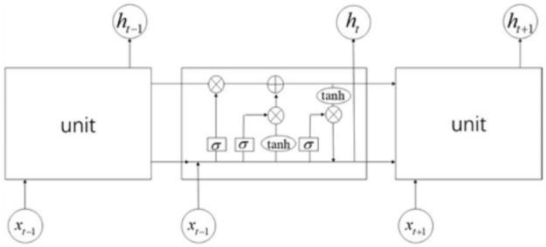

Figure3.The Operational Principle of LSTM Recurrent Neural Networks

In this context, $X(t)$ denotes the input value at time t, $h(t)$ represents the output value at time t, and C_t signifies the output state at time t. The calculation process for each logic gate follows the training sequence as outlined below:

$$i(t) = \text{sigmoid}\big(w(i) * [x(i), h(i-1)] + b(i)\big) \tag{1}$$
$$C(i) = \tanh\big(w(c) * [x(i), h(i-1)] + b(i)\big) \tag{2}$$

Equation (1) depicts the output value at time t, while Equation (2) illustrates the output state at time t. The term $i(t)$ determines whether $C(i)$ can be integrated into the state during this time period.

$$f(t) = \text{sigmoid}\big(w(f) * [x(t), h(t-1)] + b(f)\big) \tag{3}$$
$$C(t) = i(t) * C(t) + f(t) * C(t-1) \tag{4}$$

Equation (3) represents the value of the forget gate. The retention of this state in the neural network is determined by the value of $f(t)$.

$$O(t) = \text{sigmoid}\big(W(o) * [x(t), h(t-1)] + b(O)\big) \tag{5}$$
$$h(t) = O(t) * \tanh\big(C(t)\big) \tag{6}$$

Equation (5) represents the value of the output gate, which determines the output of the neural network at time t.

When the input to the input gate is of high significance to the primary state, the input gate writes this data into the neural network’ s state based on its importance. For the forget gate, if the input data significantly deviates from the network's state, the forget gate makes minor adjustments to the state and incorporates the input data according to its importance. The output of the output gate is influenced by both the most recent input data and the state of the neural network. These three gates—input, forget, and output—regulate the neural network’ s capacity to retain or discard data. This mechanism not only endows the network with the ability to remember sequential data but also prevents gradient vanishing or gradient explosion due to large data volumes. Regarding stock data, LSTM effectively addresses the challenges of prediction and handling long data sequences in stock prediction systems.

1. Experimental design

3.1. Building the LSTM Stock Prediction Model

Using the Windows 10 system, we chose to utilize CUDA for parallel computing, allowing the GPU to perform internal parallel operations and tackle many complex computational problems. The GPU's integration of high-throughput processors and computing units gives it a significant advantage in parallel computation, making it highly suitable for handling large-scale computations in neural networks. For our model, we adopted the Sequential model. This model constructs layers in a stack, processing them sequentially, ensuring connections between inputs and outputs, and maintaining adjacent relationships between layers. This approach is particularly suitable for sequential data.

In our experiment, we employed the Sequential model to construct and train single-layer, two-layer, and three-layer hidden layers on the same stock. Given that stock fluctuations are not particularly severe, this experiment uses an LSTM with fully connected layers. We input nearly two years of stock data into the neural network, including eight input features: opening price, closing price, highest price, lowest price, trading volume, turnover, and turnover rate. Each training session spans 28 days, using the second day's opening price, closing price, highest price, lowest price, trading volume, and turnover as labels. These labels are used to adjust the neural network's output loss. To optimize the model and prevent overfitting, we utilized the Adam optimizer, and applied L2 regularization and dropout mechanisms to enhance the model's generalization ability.

3.2. Experiment and Analysis

The experimental process consisted of the following steps: data acquisition, data preprocessing, noise reduction, neural network model training, hyperparameter tuning, and stock prediction.

For data acquisition, we used Python to scrape sequential data from the HIK Vision website for the years 2020 to 2023. This data included the opening price, closing price, highest price, lowest price, trading volume, turnover, and turnover rate.

In terms of data preprocessing, we addressed issues of disordered or missing values in the raw data by using interpolation and sorting operations, resulting in ten years of sequential data for HIK Vision stock. The dataset comprised over 600 entries, including data on the opening price, closing price, highest price, lowest price, trading volume, turnover, and turnover rate. For noise reduction, wavelet denoising was applied to minimize data noise.

For hyperparameter tuning, the experiment involved continuous training and parameter adjustments. Each adjustment was compared with the highest-performing parameters from previous iterations to select the optimal experimental parameters. The process is depicted in Figure 4.

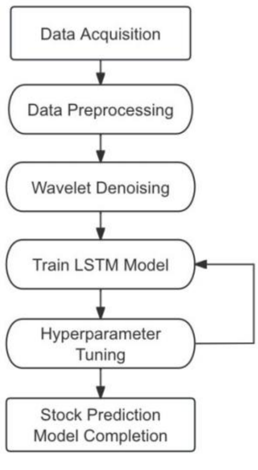

Figure4.Experimental Workflow Diagram

(1) In the first experiment, we used an LSTM neural network model with a single hidden layer to predict stock prices. Data from HIK Vision from 2018 to 2020 was input into the neural network for training.

The input layer had eight neurons, the hidden layer had 64 neurons, and the output layer had eight neurons. The stock data was trained in 28-day intervals, using the opening price, closing price, highest price, lowest price, trading volume, turnover, and turnover rate as inputs. The experiment extracted a loss value every four days, with the loss function trend shown in Figure 5.

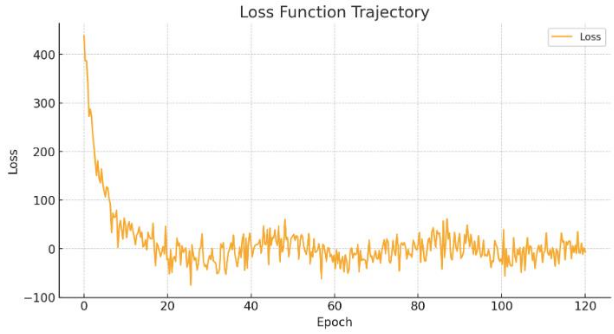

Figure5.Loss Function Trajectory

When comparing the actual closing prices with the predicted closing prices, the trajectory is shown in Figure 6:

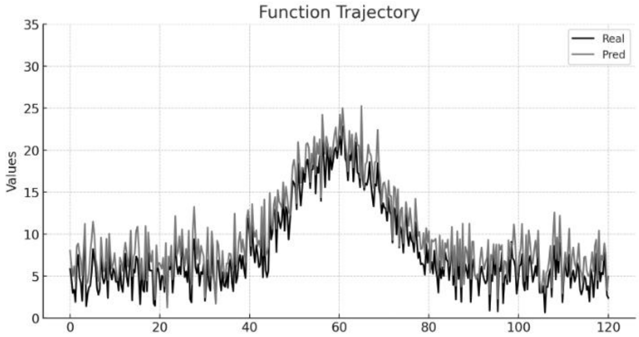

Figure6.The Trajectory of Actual Highs and Predicted Highs

It can be concluded that as the number of training iterations increases, the loss value of the LSTM neural network continues to decrease, and the actual values and predicted values converge more closely.
Next, we set up a model with two hidden layers of LSTM neural networks and one fully connected layer. Each hidden layer contains 64 neurons. Using the same input and test values as in the first experiment, the results are shown in Figure 7:

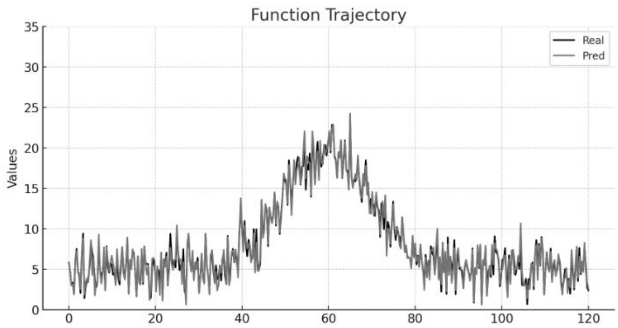

Figure7.The Trajectory of Actual Highs and Predicted Highs

The results of the second experiment indicate a significant improvement in the predictive performance of the two-layer LSTM neural network model. The speed of value fitting in Figure 7 is higher than that in Figure 6, and the actual and predicted values in Figure 7 are much closer. This demonstrates the superiority of LSTM neural networks for handling sequential data like stock prices.

The experiment reveals that increasing the number of hidden layers enhances the model′s accuracy in capturing data trends. However, when the hidden layers were increased to four, the prediction accuracy improved by only 0.02%, while adding considerable computational redundancy. Further increasing the number of layers results in negligible improvements in prediction accuracy. Therefore, the experiment recommends using a three-layer LSTM network model for stock prediction, as it strikes a good balance between prediction accuracy and computational efficiency.

4. Conclusion

This research successfully enhanced the adaptability of the LSTM neural network for stock prediction, achieving significant improvements in accuracy through innovative methods such as wavelet denoising, L2 regularization, and dropout. The fine-tuning of hyperparameters and the strategic adjustment of neurons in the hidden layers have allowed us to closely align the predicted values of HIK Vision stock trends with actual data, marking a substantial advancement in predictive proficiency. While the study faced challenges, such as the high computational demands of the LSTM model and its sensitivity to the quality of input data, these do not overshadow the achievements. The model's capacity to integrate and analyze complex data sets showcases its potential as a powerful tool in financial forecasting. Looking ahead, we are excited to explore further enhancements to our model. We plan to incorporate real-time data feeds to improve responsiveness to market changes and to expand the diversity of stocks analyzed by including various sectors and regions. This expansion will help in testing the model's robustness and generalizability. Additionally, investigating hybrid models that combine LSTM with other neural network architectures could offer new ways to balance accuracy with computational efficiency. Our ongoing efforts aim to refine our predictive framework continuously, making it an increasingly reliable and insightful tool for investors navigating the complexities of the stock market. This work not only contributes to the academic field of financial forecasting but also provides practical tools that can help investors make more informed decisions.

References

[1] Gurav, U., & Sidnal, N. (2018). Predict stock market behavior: role of machine learning algorithms. In Intelligent Computing and Information and Communication: Proceedings of 2nd International Conference, ICICC 2017 (pp. 383-394). Springer Singapore.

[2] Liu, Z., Wu, M., Peng, B., Liu, Y., Peng, Q., & Zou, C. (2023, July). Calibration Learning for Few-shot Novel Product Description. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1864-1868).

[3] Hong, S. (2021, December). Multi-task Learning Based on Multiple Data Sources for Cancer Detection. In 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 486-491). IEEE.

[4] Zi, Y., Wang, Q., Gao, Z., Cheng, X., & Mei, T. (2024). Research on the application of deep learning in medical image segmentation and 3d reconstruction. Academic Journal of Science and Technology, 10(2), 8-12.

[5] Wang, W., Gao, M., Xiao, M., Yan, X., & Li, Y. (2024). Breast Cancer Image Classification Method Based on Deep Transfer Learning. arXiv preprint arXiv:2404.09226.

[6] Hu, Y., Hu, J., Xu, T., Zhang, B., Yuan, J., & Deng, H. (2024). Research on early warning model of cardiovascular disease based on computer deep learning. arXiv preprint arXiv:2406.08864

[7] Xiao, M., Li, Y., Yan, X., Gao, M., & Wang, W. (2024). Convolutional neural network classification of cancer cytopathology images: taking breast cancer as an example. arXiv preprint arXiv:2404.08279.

[8] Song, J., & Liu, Z. (2021, November). Comparison of Norm-Based Feature Selection Methods on Biological Omics Data. In Proceedings of the 5th International Conference on Advances in Image Processing (pp. 109-112).

[9] Sun, D., Liang, Y., Yang, Y., Ma, Y., Zhan, Q., & Gao, E. (2024). Research on optimization of natural language processing model based on multimodal deep learning. arXiv preprint arXiv:2406.08838

[10] Mei, T., Zi, Y., Cheng, X., Gao, Z., Wang, Q., & Yang, H. (2024). Efficiency optimization of large-scale language models based on deep learning in natural language processing tasks. arXiv preprint arXiv:2405.11704.

[11] Saad, E. W., Prokhorov, D. V., & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. IEEE Transactions on neural networks, 9(6), 1456-1470.

[12] Schmidt, R. M. (2019). Recurrent neural networks (rnns): A gentle introduction and overview. arXiv preprint arXiv:1912.05911.

[13] Yao, J., Wu, T., & Zhang, X. (2023). Improving depth gradient continuity in transformers: A comparative study on monocular depth estimation with cnn. arXiv preprint arXiv:2308.08333.

[14] Yan, X., Wang, W., Xiao, M., Li, Y., & Gao, M. (2024). Survival Prediction Across Diverse Cancer Types Using Neural Networks. arXiv preprint arXiv:2404.08713.

[15] Liu, Z., Xia, X., Zhang, H., & Xie, Z. (2021, May). Analyze the impact of the epidemic on New York taxis by machine learning algorithms and recommendations for optimal prediction algorithms. In Proceedings of the 2021 3rd International Conference on Robotics Systems and Automation Engineering (pp. 46-52).

[16] Hong, S., He, Y., Zhang, J., Jiang, C., & Deng, Y. (2022, April). Remaining Useful Life Prediction via Bayesian Temporal Convolutional Networks. In 2021 International Conference on Advanced Computing and Endogenous Security (pp. 1-8). IEEE.

[17] Liu, H., Li, I., Liang, Y., Sun, D., Yang, Y., & Yang, H. (2024). Research on deep learning model of feature extraction based on convolutional neural network. arXiv preprint arXiv:2406.08837