A Hybrid CNN-LSTM Model for Stock Price Prediction: Enhancing Performance over Traditional Methods and Standalone LSTM

Yusheng Zhou
University of Michigan
Data Science
yszhou@umich.edu

Zhe Li
University of Michigan
Data Science
lzzzzz@umich.edu

Haoran Zhang
University of Michigan
Computer Science
zhhaoran@umich.edu

Abstract

Stock price prediction is a complex and challenging task due to the volatile and nonlinear nature of financial markets. Traditional machine learning methods like Random Forests and Support Vector Machines (SVM), as well as deep learning models like Long Short-Term Memory networks (LSTM), have been widely used but often face limitations in capturing both spatial and temporal dependencies inherent in stock data. In this paper, we propose a novel hybrid model that combines Convolutional Neural Networks (CNN) and LSTM networks to predict stock prices. The CNN component extracts local spatial features from the time-series data, while the LSTM component captures long-term temporal dependencies. Our approach demonstrates superior performance over traditional methods and standalone LSTM models, providing more accurate and reliable predictions. We discuss the architecture, implementation details, and the theoretical underpinnings that contribute to the enhanced performance of our model.

1 Introduction

Predicting stock prices is a fundamental yet highly challenging problem in financial markets, attracting significant interest from investors, analysts, and researchers. Accurate predictions can lead to substantial financial gains, but the task is complicated by the stochastic, nonlinear, and volatile nature of stock markets. Traditional statistical and machine learning methods often struggle to model these complexities effectively.

Machine learning techniques, including Random Forests and Support Vector Machines (SVM), have been applied to stock price prediction due to their ability to model nonlinear relationships. However, these models typically treat observations as independent and identically distributed, failing to capture temporal dependencies that are crucial in time-series data.

Deep learning models, particularly Recurrent Neural Networks (RNN) like Long Short-Term Memory networks (LSTM), have shown promise in modeling sequential data by capturing temporal dependencies. LSTM networks have been widely used for stock prediction tasks, leveraging their ability to remember long-term patterns. However, LSTM models may not fully exploit local spatial features present in the data, such as short-term fluctuations and local trends.

In this paper, we introduce a hybrid CNN-LSTM model for stock price prediction. The CNN component effectively extracts local spatial features from the time-series data, which are then fed into the LSTM component to capture long-term temporal dependencies. Our model aims to overcome the limitations of both traditional machine learning models and standalone LSTM models by combining their strengths.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Contributions of this paper include:

- Proposing a novel hybrid CNN-LSTM architecture tailored for stock price prediction.
- Providing a comprehensive comparison with traditional machine learning models (Random Forests and SVMs) and standalone LSTM models, highlighting the advantages of our approach.

The remainder of this paper is organized as follows: Section 2 reviews related work in stock price prediction using machine learning, LSTM models, and hybrid models. Section 3 details the methodology, including data preprocessing, model architecture, and theoretical justification. Section 4 presents the experimental setup and results. Finally, Section 5 concludes the paper and suggests directions for future research.

2 Related Work

There are many methods for stock price prediction, which can be roughly divided into two categories, statistical methods like traditional machine learning and deep learning methods. Statistical methods include logistic regression model, support vector machines, etc. Deep learning methods include multi-layer perceptron, convolutional neural network, recurrent neural network, LSTM, etc.

We reviewed ten related articles. Seven of the ten reviewed articles used artificial neural networks (ANN) [1][2][3][5][6][8][10], including three with long short-term memory (LSTM) models [3][5][6] and two with convolutional neural networks (CNN) [2][3]. One article applied support vector regression (SVR) [4], one used the autoregressive integrated moving average (ARIMA) model [7], and one applied logistic regression (LR) [9].

Traditional machine learning method or statistical methods have one big advantage that they tend to require fewer computational resources and are easy to tune. However, they perform poorly in capturing dependencies and learning the intricacies of time series data. Take SVM as an example, it's effective in high-dimensional spaces and are versatile with different kernel functions. However, its performance is poor in capturing temporal dependencies, probably for the reason that SVM treats observations as independent and identically distributed.

Deep learning has the ability of capturing complex, non-linear patterns and temporal dependencies, which traditional methods often miss. However, it also has some inevitable limitations. Firstly, the time and consumption for training a deep learning model are much higher than those of traditional methods. Secondly, due to the complexity of the stock market, even deep learning method struggles to handle all information effectively. Taking the commonly used method LSTM as an example, while it performs well in capturing long-term dependencies with its gate units, it fails in making price predictions with external factors such as news and market sentiment. Additionally, LSTMs face challenges in learning correlations over extremely long time horizons, which may cause failure in capturing the market's long-term cyclical patterns.

3 Dataset and features

3.1 Data Collection

The data utilized in this project comes from multiple sources, including Yahoo Finance , Tradingeconomics, and Investing. (https://finance.yahoo.com/, https://tradingeconomics.com/stocks, https://www.investing.com/stock-screener, https://www.investing.com/stock-screener). A Python library called yfinance was employed to retrieve the relevant data from Yahoo Finance

For numerical data, we obtained stock prices and related datasets from Yahoo Finance and performed basic data cleaning. For textual data, we scraped information from relevant websites and categorized it for storage.

Stock prices and related information from early 2018 to early 2023 were used for the experiments. The first 70 $\%$ of the data set was utilized for training the model, the following 20$\%$ was used for validation, and the remaining 10 $\%$ was reserved for final model testing.

Currently, five time series features were utilized to predict the closing prices, which are: opening price, closing price, highest price of the day, lowest price of the day, and volume for each day.

3.2 Data Preprocessing

Two different method is utilized for experiment which is random forest and CNN-LSTM.

For random forest, the training and testing input dataset features are the following data of the past 5 days: price change rate, opening price, closing price, highest price, lowest price, volume, average price of the past 10 days and 50 days. The output feature is the price change rate of the target date.

As for CNN-LSTM, one of its advantage is the ability of learning the correlation within data of time series autonomously. Therefore, the five time series features are directly inputted into the model for training.

Additionally, to avoid other interference factors, the data was normalized before the relevant training processes. After all the training was completed, the actual results were obtained through an inverse normalization process for later operation.

4 Methodology

4.1 Random forest

Random Forest is a classic Bagging model, where the weak learner is the decision tree model,which we have learned during class. Random Forest models use bootstrap to randomly take data and features to create several different sample datasets, and then build multiple decision tree models based on them. For random forest regressor, which we will use in this part of our project, the result will be generated base on the average of decision trees.

The lost function of the regressor is

$$MSE = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i - \frac{1}{n_{tree}} \sum_{j=1}^{n_{tree}} T_j(x_i))^2$$

in which n_{test} is the number of test cases, n_{tree} is the number of decision tree, T_j is the jth decision tree trained.

4.2 CNN-LSTM

Our proposed model combines CNN and LSTM layers to exploit both spatial and temporal features.

4.2.1 Convolutional Neural Network (CNN) Component

Description: CNN is a deep learning model designed for processing grid-like data such as images. It uses convolutional layers to extract features, pooling layers to reduce dimensionality, and fully connected layers for classification or regression. In our model, it's utilized to extract local spatial features from the input sequences which makes preparation for the following LSTM.

Mechanism:

- A Convolutional Neural Network (CNN) processes input data, such as images, through a series of convolutional and pooling layers to extract hierarchical features.
- In the convolutional layer, filters slide over the input to compute feature maps that highlight important patterns like edges and textures.

  $$y_{ij}^{(k)} = \sigma \left(\sum_m W_m^{(k)} * x^{(m)} + b^{(k)} \right)$$

  where $*$ denotes the convolution operation.

- The pooling layer reduces the spatial dimensions of feature maps by taking maximum or average values within a sliding window. For max pooling:

$$y_{ij} = \max_{(p,q)\in \text{window}} x_{p,q}$$

- Finally, fully connected layers process the flattened feature maps for classification or regression tasks.

$$\hat{y} = \text{ReLU}(W \cdot x + b)$$

4.2.2 Long Short-Term Memory (LSTM) Component

Description: The RNN architecture is commonly used to handle prediction problems with time series . However, it has the problem of gradient explosion and gradient vanishing, which leads to their failure in accurately capturing long-term dependencies. LSTM mitigates these issues by introducing gated units.

LSTM controls the update of cell states through three gated units.

- **Memory Cells and Gates:** Control the flow of information.
- **Forget Gate:** Decides what information to discard.
- **Input Gate:** Decides what new information to store.
- **Output Gate:** Decides what information to output.

The detailed explanation of mechanism and relevant mathematical formulas are as follows shown below.

Mechanism:

- For each timestamp, the following three steps is performed to update the cell state.
- Firstly f_t is calculated with the input and hidden state to decide how much information is reserved from the previous cell state. This step is known as forget gate.

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

- For input gate, i_t was calculated to represent how much of the candidate cell state $\tilde{C}_t$ should be added to the current cell state. Thus, the updated cell state C_t can be calculated.

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

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

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

- In the last step which is known of as the output gate, o_t was calculated to represent the activation and the hidden state is updated.

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

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

where σ is the sigmoid function, h_t and C_t represent the hidden state and cell state for timestamp t respectively.

4.2.3 Fully Connected Layer

Objective: Map the LSTM output to the final prediction.

Activation Function:

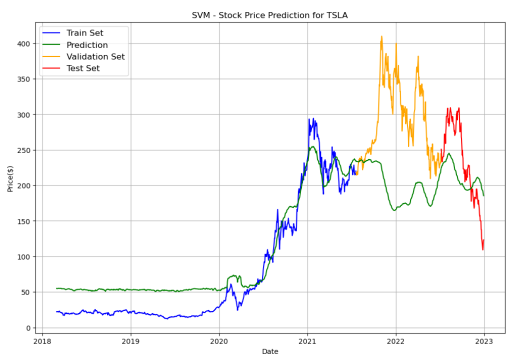
- **Regression Task:** Linear activation.
- **Classification Task:** Sigmoid or Softmax activation.

5 Experiments

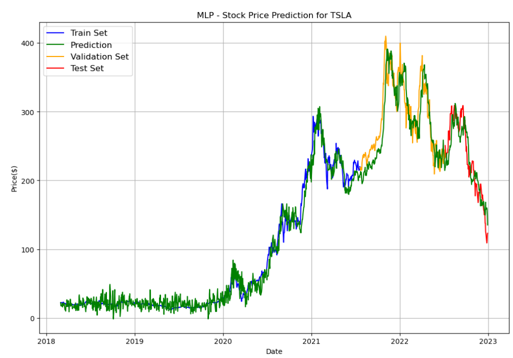
To evaluate the performance and generalizability of our proposed CNN-LSTM architecture for stock market forecasting, we conducted experiments on four randomly selected stocks: Microsoft (MSFT), Apple (AAPL), Facebook (META), and Tesla (TSLA). These stocks were chosen to capture diverse market behaviors, ensuring a robust evaluation of the model. The dataset includes historical stock prices from the past five years, incorporating features such as opening price, closing price, volume, and moving averages.

5.1 Baseline Comparison

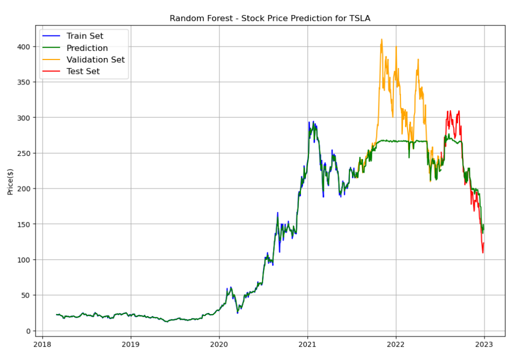
We compared the CNN-LSTM model with traditional machine learning models, including Random Forest, Support Vector Machine (SVM), and Multilayer Perceptron (MLP), which are commonly used for time series forecasting. For this comparison, we applied each model to TSLA stock data and visualized the resulting predictions, as shown below. The plots reveal that the SVM regression model



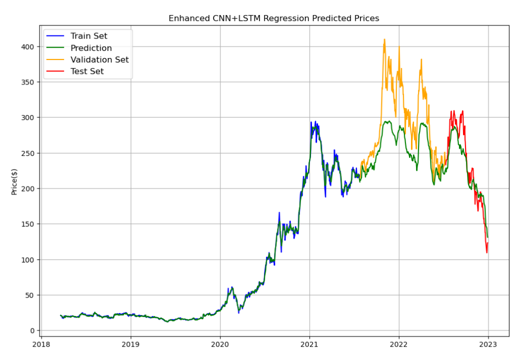
(a) SVM



(b) MLP



(c) Random Forest



(d) CNN-LSTM

Figure 1: Price prediction using different models: SVM, MLP, Random Forest, and CNN-LSTM.

underfits stock prices, while MLP tends to overfit and introduce noise, reducing prediction accuracy. In contrast, compared to random forest, which may treat stock price trends in 2022 as “noise,” our model effectively captures complex patterns, resulting in more accurate price predictions.

5.2 Ablation Experiment

To further assess the contributions of each component in our CNN-LSTM model, we conducted an Ablation Experiment by testing the following configurations:

CNN Only to observe how well CNN alone could capture short-term patterns and fluctuations in the stock prices;LSTM Only to evaluate the ability of LSTM to capture long-term dependencies in the stock market data without the help of feature extraction from CNN and CNN-LSTM (Full Model).

Below in Figure, we can easily compare with, the model using only CNN focus too much on the pattern of the price trend, making the whole graph noisy, the lstm model based too much on the ime sequence and is lag out the trend every time, which reduces the accuracy.

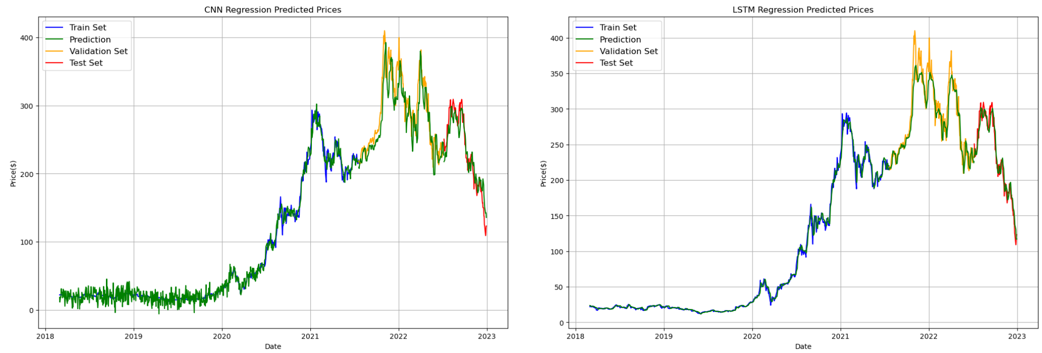
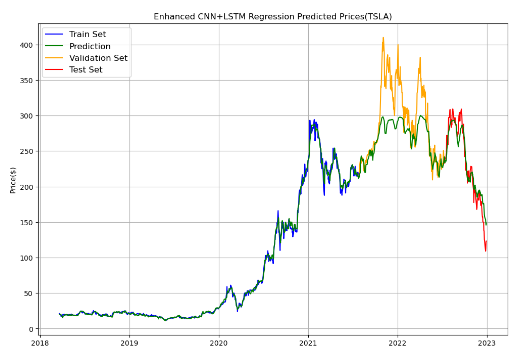


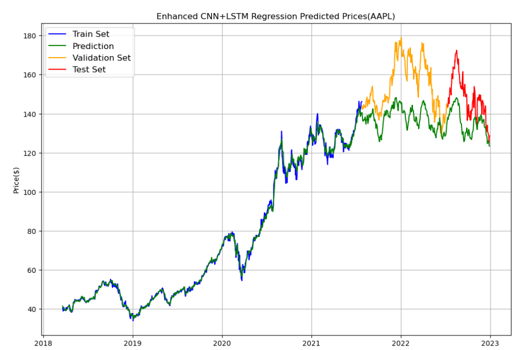
Figure 2: Price prediction using only CNN or LSTM

5.3 Model Robustness

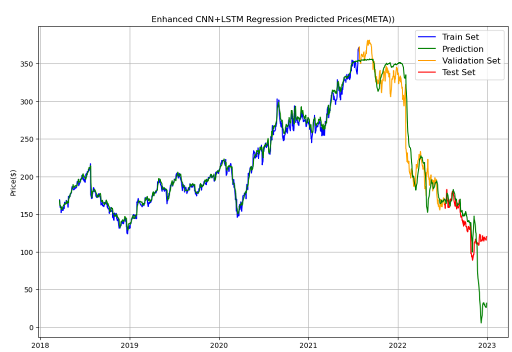
To demonstrate the robustness of our model across various stocks, we tested it using data from multiple companies, including TSLA, AAPL, META, and MSFT. Below, we present the corresponding plots for these stocks.



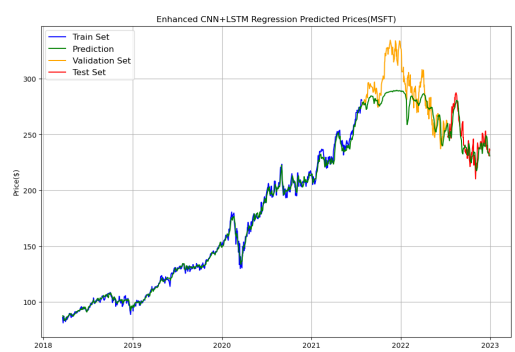
(a) TSLA



(b) AAPL



(c) META



(d) MSFT

Figure 3: Price prediction with CNN-LSTM for different stocks: TSLA, AAPL, META, and MSFT.

6 Conclusion

6.1 Result And Comparison

To analyze the performance of the CNN-LSTM model against traditional machine learning methods, we compared their accuracy rates in predicting stock price fluctuations. The accuracy rate is defined as the percentage of times the model correctly predicts the direction of stock price movements. We conducted 10 rounds of experiments for each stock and calculated the average accuracy rate.

Table 1 presents the accuracy rates in our baseline experiment, comparing the CNN-LSTM model with traditional machine learning methods such as Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Random Forest. It is evident from the results that the CNN-LSTM model demonstrates a significant improvement in prediction accuracy over the other models across all datasets.

Table 1: Accuracy rate in baseline experiment

Method|Company	AAPL	MSFT	META	TSLA
SVM	51.24	47.54	48.36	54.92
MLP	43.80	50.82	47.54	50.82
Random forest	48.49	52.92	49.18	53.74
CNN-LSTM	**53.22**	**55.32**	**52.97**	**58.34**

In our ablation study, we investigated the performance of the CNN and LSTM components individually to understand their contributions to the combined CNN-LSTM model. The results, as shown in Table 2, reveal that the CNN-LSTM model outperforms both the standalone CNN and LSTM models. This indicates that combining the two methods results in a more robust model, leveraging the feature extraction capabilities of CNN and the sequence learning strengths of LSTM.

Table 2: Accuracy rate in baseline experiment

Method|Company	AAPL	MSFT	META	TSLA
CNN(only)	46.43	49.33	50.24	54.27
LSTM(only)	43.18	53.04	49.87	54.67
CNN-LSTM	**53.22**	**55.32**	**52.97**	**58.34**

In conclusion, our experimental results clearly demonstrate the superiority of the CNN-LSTM model over traditional machine learning methods for predicting stock price fluctuations. The combined model has shown notable improvements in accuracy across multiple stock datasets, highlighting its effectiveness in handling temporal and spatial features in stock market data. Additionally, the ablation study confirms that the integration of CNN and LSTM components leads to enhanced performance, validating the design choice of using a hybrid architecture.

6.2 Summary

In our project, we developed a hybrid CNN-LSTM model for predicting stock prices. Utilizing the yfinance library as our data source, we trained our model on the historical stock prices of Apple Inc, Microsoft, Meta (Facebook), and Tesla to obtain predicted results for stock price movements. Additionally, we constructed traditional models such as random forest for comparison.

The results indicate that our model successfully captures the basic trends of stock prices, outperforming traditional methods in accuracy. However, there is room for improvement in enhancing the precision of the predicted price change rates.

6.3 Advantages

Based on the result of our hybrid CNN-LSTM model and the traditional model, we can conclude that the average accuracy of our model is higher then the average accuracy of the traditional model. Also, the accuracy fo the prediction of rise or fall is stably higher than 50%, illustrating that despite of the large error, the model does provide some information about the trend of the stock price. The small distance between the predicted price curve and the actual price curve also shows that the model is generally correct.

6.4 Disadvantages

Comparing to the traditional model, our hybrid CNN-LSTM model needs more time to complete training and predicting (the average time needed for CNN-LSTM model is about 80s, while random forest only need about 3 seconds). Some other aspect includes more parameters, including the batch-size, time step, number of the layers to adjust, which increase the difficulty to improve the performance. Also, ignorance of aspect except the historical data (like the news) may be a major cause to the low precision.

6.5 Future improvement

Some of the possible improvement that can be applied to our model include following aspects:

Adjusting parameters: by adjusting the parameters including the length of time step, the number of layers and batch-size. One possible way is to apply different value of parameters to the model and select the ones with the highest average accuracy

Considering more non-statistic variable: stock price may be influenced by a lot of non-statistic variables (i.e. news of the company, economic environment). We can add these variant into consideration to adjust our model. One of the way is analyzing the new about the company using algorithm similar to the one we used to classify spam in our homework. We can classify news into good news and bad news to analyze its influence to the stock price.

7 Contributions

This project was completed with the effort of the whole team. The individual contribution of each member are presented below.

Zhe Li: At the early stage of the project, Zhe Li was responsible for literature review. At the implementation stage of the project, he carried out the experiment of using Random forest and wrote relevant part in the report.

Haoran Zhang: At the early stage of the project, Haoran Zhang was responsible for coming up with a new idea based on the literature review. Additionally, he carried out the experiment of using CNN-LSTM and wrote the experiment part in the report.

Yusheng Zhou: At the early stage of the project, Yusheng Zhou collected all the data utilized in this project and wrote the proposal. He also took responsibility of the experiment of using LSTM and wrote the relevant part in the report.

References

[1] Singh, R. & Srivastava, S. (2017) Stock prediction using deep learning. *Multimed Tools Appl*. 76:18569-18564.

[2] Chen, S. & He, H. (2018) Stock Prediction Using Convolutional Neural Network. *IOP Conference Series: Materials Science and Engineering.* (435)012026.

[3] Lu, W., Li, J. & Wang, J. (2021) A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications.* 33:4741-4753.

[4] Schumaker, R.P. & Chen, H. (2009) A quantitative stock prediction system based on financial news. *Information Processing and Management.* 45:571-583.

[5] Ding, G. & Qin, L. (2020) Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics.* 11:1307–1317

[6] Chen, K., Zhou, Y. & Dai, F. (2015) A LSTM-based method for stock returns prediction: A case study of China stock market. *IEEE International Conference on Big Data.*

[7] Adebiyi, A.A., Adewumi, A.O. & Ayo, C.K. (2014) Stock Price Prediction Using the ARIMA Model. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation.*

[8] Qiu, M., Song, Y. & Akagi, F. (2016) Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos, Solitons and Fractals.* (85)1-7

[9] Dutta, A. Bandopadhyay, G. & Sengupta, S. (2012) Prediction of Stock Performance in the Indian Stock Market Using Logistic Regression. *International Journal of Business and Information.* (7)105.

[10] Oliveira, F.A., Zarate, L.E.& Reis, M.A. (2011) The Use of Artificial Neural Networks in the Analysis and Prediction of Stock Prices. *IEEE International Conference on Systems, Man and Cybernetics.*

Appendix

Code Availability

The code for reproducing the results presented in this paper is available at the following GitHub repository: https://github.com/zhhr0321/stock$_prediction - stats415$.