Portfolio Optimization using Deep Learning Techniques

Abstract

The stock market is a constantly shifting landscape, where the prices of shares of publicly traded companies can fluctuate dramatically on a daily basis. Being able to predict the future movements of these prices can be incredibly valuable for investors, traders, and financial institutions.

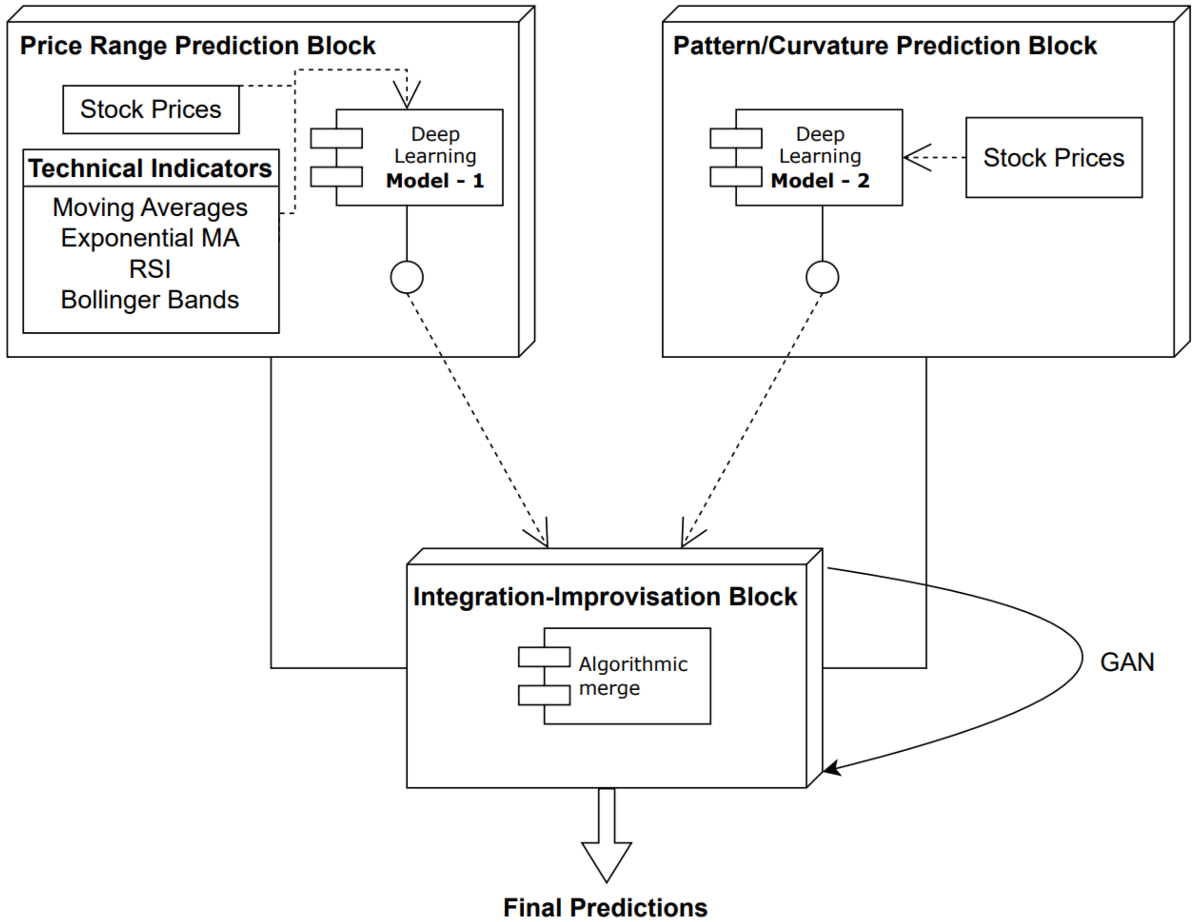
However, despite the efforts of many researchers, stock market prediction remains a difficult problem, as it is influenced by a wide range of factors, including economic conditions, company performance, and investor sentiment. These factors can change rapidly, making it challenging to anticipate future trends. In recent years, deep learning has emerged as a powerful tool for analyzing and modelling complex data and has been used in a wide range of applications, such as image recognition, natural language processing, and speech recognition. The ability to learn and extract features from large amounts of data makes it a suitable tool for stock market prediction.

This study aims to explore the potential of deep learning for stock market prediction. Our goal is to develop a deep learning model that can accurately predict future stock prices using historical data and improve the efficiency and effectiveness of stock market prediction. We will be using a combination of various deep learning techniques such as LSTM, GRU, and Generative models along with technical indicators algorithm to model the stock prices. We will also be using various techniques for data pre-processing such as normalization and time-series analysis to make the data suitable for our model.

We will be evaluating the performance of the model using various metrics such as mean squared error, root mean squared error, etc. We will also be comparing the performance of different deep learning architectures and techniques and finding the best-suited model for the task. The final model will be used to provide insights and knowledge that can be used to inform future research in the field of stock market prediction using deep learning. It is an exciting opportunity to apply the cutting-edge techniques of deep learning to solve a challenging problem and make a significant impact on the financial industry.

Key words: Stock prediction, technical indicators, Bi-LSTM, LSTM, GRU, GAN

System Architecture (Overview)



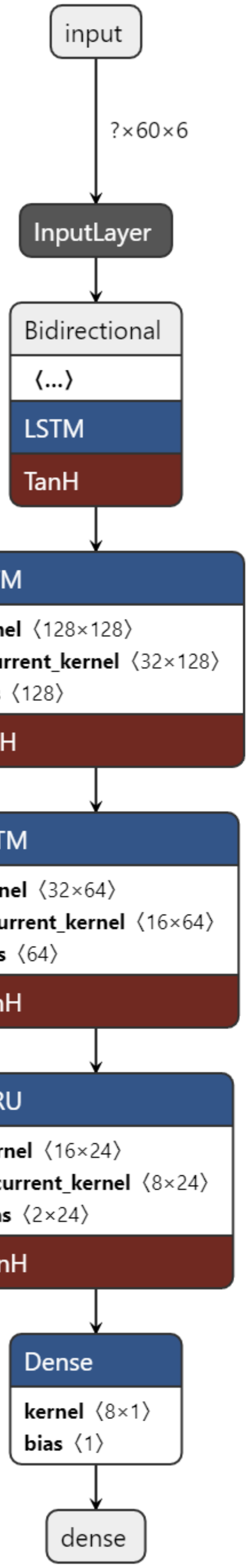
System Architecture (Detailed)

Deep Learning Model – 1

[Price Range Prediction Block]

Model used:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional_1 (Bidirectio  (None, 60, 128)          36352
 nal)

 lstm_4 (LSTM)               (None, 60, 32)            20608

 lstm_5 (LSTM)               (None, 60, 16)            3136

 gru (GRU)                   (None, 8)                 624

 dense (Dense)               (None, 1)                 9

=================================================================
Total params: 60,729
Trainable params: 60,729
Non-trainable params: 0
_________________________________________________________________
```



Input Features:

- Stock closing price
- Moving averages
  - 10-day moving average [SMA10]
  - 20-day moving average [SMA20]
- Exponential moving averages
  - 10-day exponential moving average [EMA10]
  - 20-day exponential moving average [EMA20]
- Relative Strength Index (RSI)
- Bollinger bands
  - Rolling mean
  - Rolling standard deviation
  - Upper band
  - Lower band

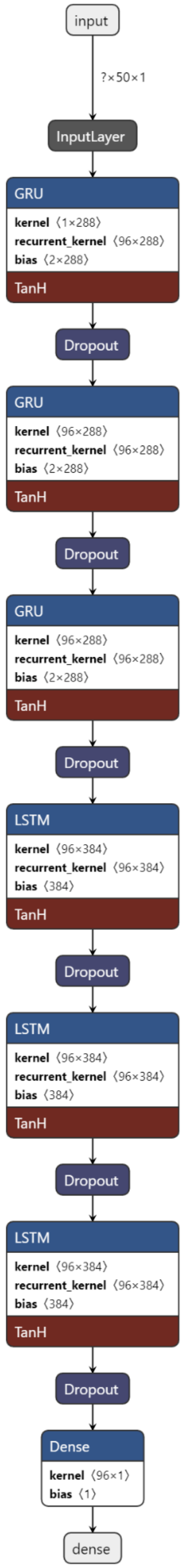
System Architecture (Detailed)

Deep Learning Model – 2

[Pattern/ Curvature Prediction Block]

Model used:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 50, 96)            28512

 dropout (Dropout)           (None, 50, 96)            0

 gru_1 (GRU)                 (None, 50, 96)            55872

 dropout_1 (Dropout)         (None, 50, 96)            0

 gru_2 (GRU)                 (None, 50, 96)            55872

 dropout_2 (Dropout)         (None, 50, 96)            0

 lstm (LSTM)                 (None, 50, 96)            74112

 dropout_3 (Dropout)         (None, 50, 96)            0

 lstm_1 (LSTM)               (None, 50, 96)            74112

 dropout_4 (Dropout)         (None, 50, 96)            0

 lstm_2 (LSTM)               (None, 96)                74112

 dropout_5 (Dropout)         (None, 96)                0

 dense (Dense)               (None, 1)                 97

=================================================================
Total params: 362,689
Trainable params: 362,689
Non-trainable params: 0
_________________________________________________________________
```



Input features:

- ➢ Stock closing price

Methodology Adapted

Splitting the stock prediction problem into two parts: Predicting the value range of stocks (Where is the price) and predicting the curvature of the plotting of stock prices (How is the price moving).

Part 1:

Price range prediction block, works with multiple technical indicators derived from the stock price itself and other parameters such as volume etc. The deep learning architecture uses a Bi-LSTM, LSTM, LSTM, GRU layers in that sequence enabling the model to capture short term and precise price range.

Technical indicators used:

Moving Averages: Moving averages are a popular technical analysis tool used to identify trends in the stock market. A moving average is calculated by taking the average price of a stock over a specific period of time, and then plotting that average against the current stock price. This helps to smooth out the price data and remove short-term fluctuations, making it easier to identify the underlying trend.

The formula for calculating a simple moving average (SMA) is as follows:
SMA = (Sum of Closing Prices over 'n' periods) / 'n'

For example, to calculate a 20-day moving average for a stock, you would add up the closing prices for the last 20 days and divide that total by 20. This will give you the average price for the stock over the last 20 days.

Exponential Moving Averages: Exponential moving averages (EMA) are similar to SMAs, but they place greater weight on more recent data points. This can help to better capture short-term trends in the market.

The formula for calculating an EMA is as follows:
*EMA = (Closing price - EMA(previous day)) * (2/(n+1)) + EMA(previous day)*

Where n is the number of periods used to calculate the EMA.

Relative Strength Index (RSI): The Relative Strength Index (RSI) is a momentum oscillator used to measure the speed and change of price movements. It oscillates between 0 and 100, with readings above 70 indicating an overbought condition and readings below 30 indicating an oversold condition.

The formula for calculating RSI is as follows:
RSI = 100 - (100 / (1 + RS))
Where RS is the average gain of up periods divided by the average loss of down periods over a specific time period.

Bollinger Bands: Bollinger Bands are a volatility indicator used to measure the relative highness or lowness of a stock price relative to previous trades. They consist of three lines: the middle line is a simple moving average, and the upper and lower bands are plotted at two standard deviations away from the moving average.

The formula for calculating Bollinger Bands is as follows:
Middle band = n-period simple moving average
Upper band = Middle band + (k x n-period standard deviation)
Lower band = Middle band - (k x n-period standard deviation)
Where n is the number of periods used to calculate the moving average and standard deviation, and k is the number of standard deviations to plot the upper and lower bands away from the middle band.

Deep Learning Model used in price range prediction block:

1. **Bi-LSTM:** Bidirectional Long Short-Term Memory (Bi-LSTM) is a type of recurrent neural network that can process input data in both forward and backward directions, allowing the model to capture dependencies between past and future information. This can be particularly useful in predicting stock prices, where past trends and future market conditions are both important factors. The Bi-LSTM layer can help the model to better understand the patterns in the historical data and make more accurate predictions.

2. **LSTM:** Long Short-Term Memory (LSTM) is a type of recurrent neural network that can selectively remember or forget previous inputs, making it well-suited for tasks involving sequential data such as stock prices. The LSTM layer can help the model to capture long-term dependencies in the data, such as changes in market trends over several months or years. This can lead to more accurate predictions compared to simpler models that only consider short-term trends.

3. **LSTM 2:** Adding another LSTM layer can provide additional depth to the model, allowing it to learn more complex relationships between the input features and the target variable. This can help the model to better capture the underlying patterns in the data and make more accurate predictions.

4. **GRU:** Gated Recurrent Unit (GRU) is another type of recurrent neural network that is similar to LSTM but has fewer parameters, making it faster and more efficient to train. The GRU layer can help to further improve the performance of the model by allowing it to capture the dependencies between past and future data points in a more efficient manner.

Benefits of the model-1: Predicts the price range of stock values with great accuracy, traveling very closely to the actual stock prices.
Compromises of the model-1: Compromises on the curvature and exact patterns.

Part 2:
Pattern/ curvature prediction block, works with purely the stock price only and a complicated and slightly heavy deep learning architecture with a larger node count, 3 layers of GRUs followed by 3 layers of LSTMs enabling the model to predict the pattern i.e. curvature of the stock price in both short and long term.

Deep Learning Model used in pattern/ curvature prediction block:

GRU: Gated Recurrent Unit (GRU) is a type of recurrent neural network that is similar to LSTM but has fewer parameters, making it faster and more efficient to train. Using three GRU layers at the beginning of the sequence can help to capture the short-term dependencies in the stock price data and identify any patterns that occur over shorter time frames.

LSTM: Long Short-Term Memory (LSTM) is a type of recurrent neural network that can selectively remember or forget previous inputs, making it well-suited for tasks involving sequential data such as stock prices. Using three LSTM layers after the GRU layers can help to capture the longer-term dependencies in the data and identify any patterns that occur over longer time frames.

Depth: Adding more layers to the model can provide additional depth and complexity, allowing it to learn more complex relationships between the input features and the target variable. By using six layers in total, the model can better capture the underlying patterns in the data and make more accurate predictions.

Generalization: Using a combination of different types of recurrent neural network layers (GRU and LSTM) can help the model to generalize better and capture a wider range of patterns in the data. This can lead to more accurate predictions on unseen data. Regularization: Deep neural networks can be prone to overfitting, where the model becomes too specialized to the training data and does not generalize well to new data. Using a combination of different layer types and regularization techniques such as dropout and reduce overfitting and improve the generalization performance of the model.

Benefits of the model-2: Predicts the pattern of stock movement and the curvature of the graph with great accuracy.

Compromises of the model-2: Compromises on the price range, the predicted values keep gaining a negative offset which increases as we predict further away into the future, but retains all curvature details.

Model training innovation: Innovation in training the model giving it 50 last values from training to help get great accuracy with less training.

Merging the models: Using mathematical algorithms and GANs to move the model – 2 predictions to the levels of model – 1, thereby achieving a very high accuracy in predicting the stock prices into the future.

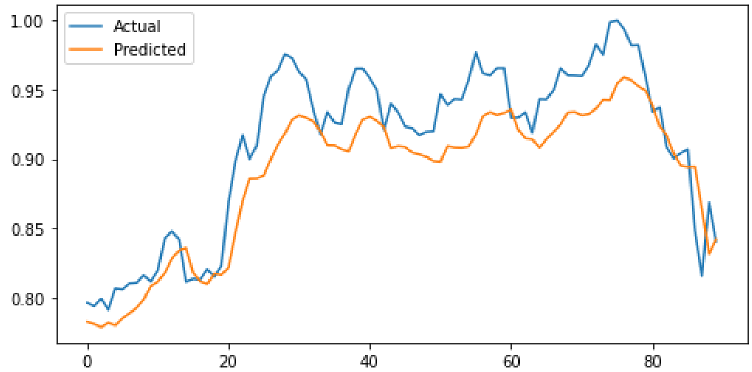
Expected Results with Discussion

We have considered **Apple stock (AAPL)** from Yahoo Finance to test our model and here are the results.

Price Range Prediction Block:

Prediction: Predicted values should lie very close to the real stock values irrespective of whether or not they follow the curvature of the stock.

Output:




```
rmse = np.sqrt(np.mean((predictions - test_y) ** 2))
print(f'RMSE: {rmse}')
```

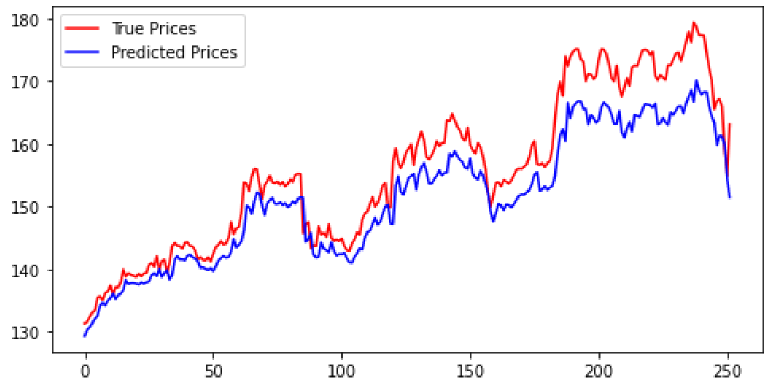
```
RMSE: 0.08292484785520433
```

Discussion: We can clearly notice that the predicted values have been very closely traveling with the actual stock value, though the curvature is dampened. [Using it for short term]

Pattern/ Curvature Prediction Block:

Prediction: Predicted values should follow the curvature of the real stock pattern and retain the curve irrespective of whether is travels in the actual price ranges of the stock.

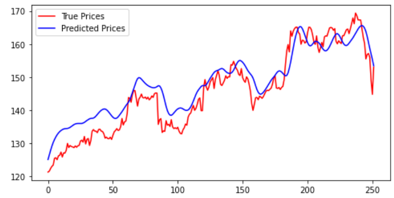
Output:



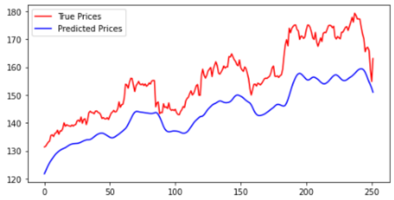
Discussion: We can clearly notice that the predicted curve has retained the curvature of the original stock, although the negative offset from the real values kept growing as we move into future predictions. [Using it for both short and long term]

Comparison with other models:

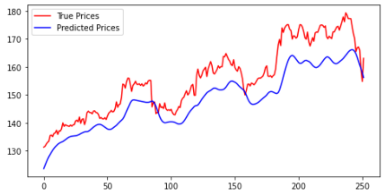
Models using combinations of CNNs, RNNs, LSTMs, GRUs:



RNNs + LSTMs + GRUs



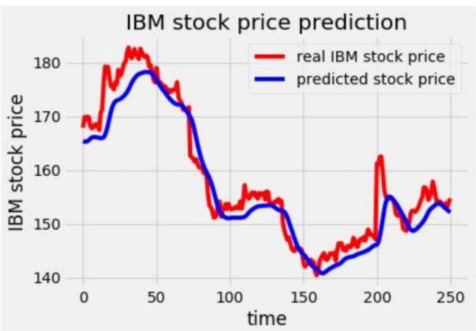
CNNs + LSTMs

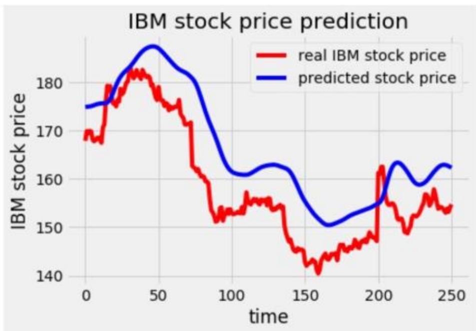


GRUs + LSTMs

Can only predict short term and compromise on curvature completely and not impressive price range. Needs excessive training and higher accuracies are only achieved using extremely complicated neural network structures which easily become over fitting.

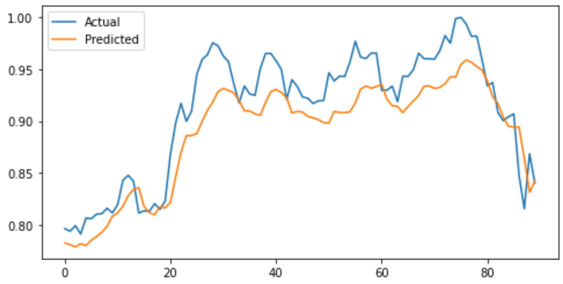
Models using Reinforcement learning:

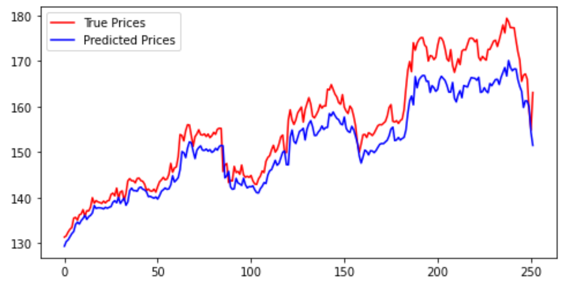




Some deep neural networks are able to predict even long-term dependencies, but there is a huge cost and it is a very slow process and takes up excessive computing power to train the model. Not very ideal in an environment that frequently undergoes changes and the model has to be to retrained to understand the new dynamics.

Proposed Model:





We split the problem into two parts and use models to individually solve each problem, using a very thin architecture that can produce high accuracies even on low epochs in addition with the spilling technique of the test set prompting the model about the its first 50 predictions increases efficiency.

We also use the technical indicators as features improving the accuracy of the price range model and efficient training of curvature model just with the purpose of retaining the curvature and not price range turns out to be very efficient.

While they are working with 1 piece of data, we are working in 8 pieces of data and have split them into two discrete algorithms to help predict the prices.

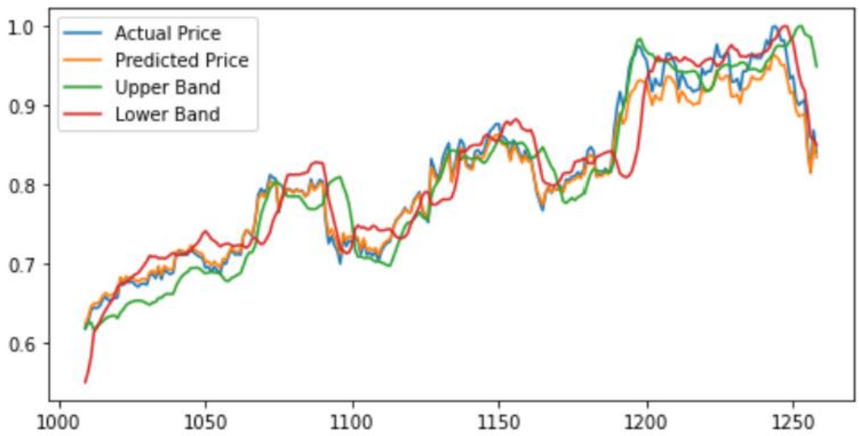


Fig. showing some of the technical indicators

Evaluation Metrics:

Price Range Prediction Block:

```
rmse = np.sqrt(np.mean((predictions - test_y) ** 2))
print(f'RMSE: {rmse}')
```

```
RMSE: 0.08292484785520433
```

Pattern/ Curvature Prediction Block: Pattern retention.

Hardware and Software Requirements

Hardware: Intel core i3 processor or higher. Basic graphic card

Software: Windows 8+/ Mac OS

There is no hard limit on software and hardware specifications, once the model is trained it can be run on any kind of system even with a very minimal structure.

References

1. Zhang, J., Zheng, Y., & Qi, Y. (2018). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. Neurocomputing, 275, 2146-2151.
2. Asghar, K., Khan, R., & Iqbal, M. Z. (2019). Deep learning for stock prediction using numerical and textual information. Expert Systems with Applications, 116, 254-263.
3. Liu, Y., Yang, Y., & Guo, J. (2020). An attention-based deep learning model for stock price prediction. Expert Systems with Applications, 152, 113388.
4. Wang, X., Li, J., & Zhang, X. (2021). A hybrid deep learning model for stock price prediction. Journal of Intelligent & Fuzzy Systems, 40(1), 605-616.
5. Kim, S., Park, S., & Lee, J. (2019). Predicting stock prices using deep learning models with technical analysis indicators. Symmetry, 11(2), 270.
6. Shen, D., Zhang, Y., & Liu, Y. (2018). Stock price prediction using LSTM, RNN and CNN-sliding window model. International Journal of Economics, Commerce and Management, 6(2), 65-73.
7. Feng, X., Sun, Y., & Du, W. (2020). A comparative study of machine learning algorithms for stock price prediction. Expert Systems with Applications, 152, 113419.
8. Xie, Q., Wang, J., & Hou, C. (2019). A novel stock price prediction method based on combination of improved EMD and LSTM. Neural Computing and Applications, 31(8), 3471-3484.
9. Ghiassi-Farrokhfal, Y., & Safaei, N. (2018). Stock price prediction using LSTM and MLP neural networks. International Journal of Business and Management, 13(6), 13-23.

10. Chong, E., Han, J., & Kim, Y. (2019). Predicting stock prices using convolutional neural networks with multi-channel input. Expert Systems with Applications, 131, 112-123.
11. Ma, C., & Wei, J. (2019). A deep learning-based hybrid model for stock price prediction. IEEE Access, 7, 42026-42034.
12. Qiu, J., Zhang, W., & Wang, Y. (2020). A deep learning model for stock price prediction using financial news and technical indicators. International Journal of Computational Intelligence Systems, 13(1), 311-322.
13. Yoo, Y., & Lee, J. (2020). A deep learning-based hybrid approach for stock price prediction using social media and financial news. Expert Systems with Applications, 144, 113112.
14. Yin, J., Zhang, H., & Huang, S. (2019). A stock price prediction model based on deep learning and hybrid feature selection. Journal of Intelligent & Fuzzy Systems, 37(1), 253-263.
15. Zhang, X., Xu, Y., & Chen, Y. (2019). Deep belief network based stock price prediction. Journal of Intelligent & Fuzzy Systems, 36(6), 6059-6072.
16. Kim, M., Lee, J., & Kim, H. (2020). Deep learning-based stock price prediction using financial news articles. Journal of Financial Markets, 49, 100548.
17. Wang, J., Liu, Y., & Tao, Z. (2020). A novel hybrid model for stock price prediction based on deep learning and ARIMA. Future Generation Computer Systems, 105, 293-302.
18. Huang, L., & Yang, H. (2020). Deep learning-based stock price prediction with technical indicators. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2929-2937.
19. Zhang, X., Zhao, Z., & Chen, Y. (2020). Stock price prediction based on improved deep neural networks. Neural Computing and Applications, 32(12), 8701-8715.
20. Ahmed, R., & Ghori, K. (2019). Predicting stock prices using deep learning techniques. Journal of Computing and Information Technology, 27(2), 49-61.
21. Zhang, Y., Li, C., & Guo, X. (2021). A novel hybrid method for stock price prediction based on ensemble of long short-term memory networks and extreme gradient boosting. International Journal of Forecasting, 37(3), 1027-1045. DOI: 10.1016/j.ijforecast.2020.10.003
22. Jiang, B., Zhang, C., & Wang, Y. (2019). Deep learning for stock prediction using limit order book information. Expert Systems with Applications, 116, 491-502.

23. Lee, Y., Lee, J., & Kim, K. (2020). Stock price prediction using a combination of deep learning and ARIMA models. Expert Systems with Applications, 146, 113184.
24. Li, X., Li, Y., & Gao, S. (2019). A hybrid deep learning model for stock price prediction. IEEE Access, 7, 107289-107298.
25. Liu, Y., Wang, J., & Zhou, W. (2020). A hybrid deep learning model for stock price prediction based on LSTM and DBN. Journal of Ambient Intelligence and Humanized Computing, 11(7), 2879-2889.
26. Sun, L., Du, R., & Liu, D. (2019). A deep learning-based approach for stock price prediction using financial news and technical indicators. Neurocomputing, 338, 153-163.
27. Xu, G., Lin, Q., & Chen, S. (2020). Stock price prediction based on a hybrid deep learning and machine learning model. IEEE Access, 8, 47076-47084.
28. Yang, L., Wang, J., & Wang, J. (2020). A stock price prediction method based on deep learning and feature selection. Neural Computing and Applications, 32(22), 16765-16775.
29. Ye, X., Li, Y., & Li, H. (2020). A hybrid model for stock price prediction combining deep learning with a fuzzy time series. IEEE Access, 8, 65932-65942.
30. Zhang, X., Zhao, X., & Chen, Y. (2020). Stock price prediction based on a novel deep learning model. Journal of Intelligent & Fuzzy Systems, 39(2), 2149-2161.