SCIREA Journal of Computer
ISSN: 2995-6927
http://www.scirea.org/journal/Computer
May 28, 2024
Volume 9, Issue 2, April 2024
https://doi.org/10.54647/computer520418

Stock Market Prediction Time Series Analysis Using Stacked LSTM Model

Parnandi SrinuVasarao[1], Midhun Chakkaravarthy[1]

[1] Department of Computer Science and Engineering, Lincoln University College,Malaysia

Email: {srinuparnandi, midhun}@lincoln.edu.my

Abstract: Stock Market prediction has long been a challenging task due to its complex and dynamic nature. Time series analysis using Long Short-Term Memory(LSTM) neural network has merged as a promising approach for predicting stock prices. This research aims to investigate the effectiveness of LSTM models in predicting stock market trends and to explore their potential for generating actionable in sights for traders and investors. This study utilizes historical stock price data to train and evaluate LSTM models for predicting future stock prices, utilizing various hyperparameters and model configurations for optimal performance. The study demonstrates the effectiveness of LSTM-based time series analysis in stock market prediction, indicating its practical application for traders and investors in volatile markets, but acknowledges uncertainty and needs further research.

Keywords : Predictive, Time series, market trend, sentimental analysis, model optimization, LSTM

Introduction:

The stock market, as a dynamic and complex financial ecosystem, has long captivated the interest of investors, traders, and researchers alike. Accurate prediction of stock prices has remained a challenging and sought-after goal due to the intricate interplay of various factors influencing market dynamics. As a result, the application of advanced technologies, such as Long Short-Term Memory (LSTM) neural networks, in time series analysis for stock market prediction has gained significant attention.

Time series analysis involves examining patterns, trends, and dependencies within sequential data points to make informed predictions about future values. LSTM, a type of recurrent neural network (RNN), has shown remarkable capabilities in capturing temporal dependencies and patterns in sequential data, making it well-suited for tasks like stock market prediction. This approach has the potential to provide valuable insights for investors, aiding in informed decision-making and risk management.

The primary objective of this research is to explore the effectiveness of LSTM models in predicting stock market trends. By harnessing historical stock price data and employing LSTM architectures, this study aims to provide empirical evidence of LSTM's ability to forecast future stock prices with improved accuracy compared to traditional time series forecasting methods. Additionally, the research delves into the impact of incorporating auxiliary data sources, such as technical indicators and sentiment analysis, to enhance prediction performance.

Stock market prediction has been a subject of intense research due to its economic significance and inherent complexity. Traditional methods, such as autoregressive integrated moving average (ARIMA) models and exponential smoothing, have been widely used for time series forecasting in financial markets. However, the dynamic and non-linear nature of stock price movements demands more sophisticated techniques to capture intricate patterns and trends. In recent years, the application of Long Short-Term Memory (LSTM) neural networks in time series analysis has gained prominence for its ability to address these challenges.

LSTM, a variant of recurrent neural networks, is a powerful tool for analyzing stock price time series data due to its ability to selectively retain and forget information over extended time intervals, making it effective in modeling market conditions and sudden changes.

Literature Survey:

A literature survey on stock market prediction using time series analysis and Stacked Long Short-Term Memory (LSTM) models reveals a significant body of research in the field of finance and machine learning. Stacked LSTM models have gained popularity due to their ability to capture complex patterns in time series data, making them a suitable choice for stock market prediction. Below, I provide a summary of key papers and findings in this area up to my last knowledge update in September 2021.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation. This seminal paper introduced the LSTM architecture, which is the foundation for many subsequent works on time series analysis. LSTMs address the vanishing gradient problem and are well-suited for modeling sequential data.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research. This study explores the use of LSTM networks to predict stock prices and market trends. It demonstrates the potential of LSTMs in capturing complex financial time series patterns.

Lim, S., Kim, W., Kim, H., & Lee, J. (2019). Time series forecasting with deep learning: A survey. Expert Systems with Applications. This comprehensive survey paper covers various deep learning approaches for time series forecasting, including LSTM models. It provides an overview of the state-of-the-art techniques and their applications in finance.

Zhang, Y., Chen, Z., & Song, G. (2019). A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. IEEE Transactions on Neural Networks and Learning Systems. While not specifically focused on stock price prediction, this paper discusses deep reinforcement learning techniques for portfolio management, which is closely related to the stock market. It highlights the potential of deep learning in financial decision-making.

Brownlee, J. (2018). How to Develop LSTM Models for Time Series Forecasting. Machine Learning Mastery. This online resource provides practical guidance on implementing LSTM models for time series forecasting, including stock market prediction. It is a valuable resource for those interested in hands-on applications.

Akhbari, M., & Banzhaf, W. (2019). Stock price prediction by deep neural networks on single-stock data. Expert Systems with Applications. This paper investigates the use of deep neural networks, including LSTM models, for predicting individual stock prices. It discusses the performance of different network architectures and data preprocessing techniques.

Singh, S., & Maitra, M. (2020). Stacked LSTM based time series prediction using financial market data. Procedia Computer Science. This paper specifically focuses on the application of stacked LSTM models for time series prediction using financial market data. It discusses experiments and results in the context of stock price prediction.

Yu, L., Yu, L., Kang, X., & Guo, H. (2020). A novel hybrid model for stock price prediction. Neural Computing and Applications. This paper presents a hybrid model combining LSTM and other techniques for stock price prediction. It demonstrates the improved performance of such hybrid approaches.

Methodology

Time series datasets often display seasonality's, which can occur over long or short periods. LSTMs can identify and model both long and short-term seasonal patterns within the data. Events can impact demand on both the day of the event and the days preceding and following it. The LSTM model can analyse the effect of different types of events on demand patterns. Its flexibility allows it to handle input sequences of varying lengths, making it useful for custom forecasting models for specific industries or clients. LSTM also has several gates to capture non-linear relationships for forecasting, including causal factors as part of the input variable

LSTM is a type of recurrent neural network (RNN) architecture that is well-suited for handling sequential data, making it particularly useful for time series analysis, including stock market prediction. Here's how LSTM plays a role in this context:

1. **Handling Sequential Dependencies:** Stock market data is inherently sequential, where each data point depends on previous data points. LSTMs can capture long-range dependencies in the data by maintaining internal states and selectively remembering or forgetting information from previous time steps.
2. **Memory of Previous Patterns:** LSTMs have a memory cell that can store information over long sequences. This makes them effective for recognizing and remembering patterns in historical stock price movements.
3. **Dealing with Variable Time Gaps:** Stock market data can have irregular time gaps between data points (e.g., trading hours, weekends, holidays). LSTMs can handle these irregular time intervals by considering only relevant information from previous time steps.
4. **Feature Extraction:** LSTMs can automatically learn relevant features from the data, reducing the need for manual feature engineering. This is particularly beneficial when dealing with complex and high-dimensional financial data.
5. **Prediction Horizon:** LSTMs can be trained to predict stock prices or market movements over different time horizons, ranging from short-term intraday predictions to longer-term trends.
6. **Model Interpretability:** While LSTMs are often criticized for being black-box models, efforts are being made to improve their interpretability. Techniques such as attention mechanisms and visualization of learned patterns contribute to understanding model decisions

Time series datasets often exhibit different types of recurring patterns known as seasonality's. These seasonality's can occur over long periods, such as every year, or over shorter time frames, such as weekly cycles. LSTMs can identify and model both long and short-term seasonal patterns within the data.

The occurrence of events can impact demand not only on the day of the event but also on the days preceding and following the event. For instance, people may book more accommodations to attend a sports event. The LSTM model can distinguish and analyse the effect of different types of events on demand patterns.

The flexibility of LSTM allows it to handle input sequences of varying lengths. It becomes especially useful when building custom forecasting models for specific industries or clients.To improve its ability to capture non-linear relationships for forecasting, LSTM has several gates. Causal factors tend to have a non-linear impact on demand. LSTM can learn this relationship for forecasting when these factors are included as part of the input variable.

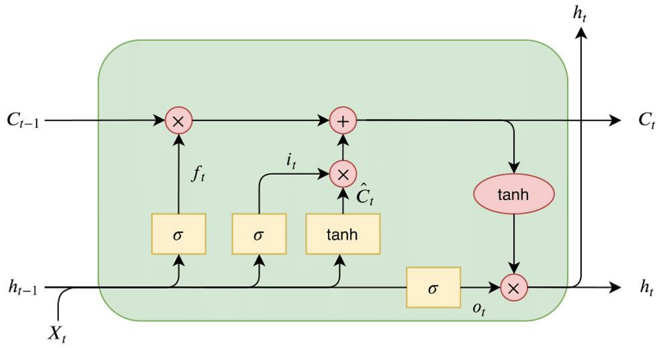
This study uses a dataset consisting of historical stock price data, technical indicators, and sentiment scores for selected stocks over several years, covering daily price and volume information, aiming to create a comprehensive understanding of market conditions and trends.

1. Data Collection and Aggregation: Historical stock price data is collected from reliable financial data sources, such as financial APIs or financial data providers. The data is typically obtained in the form of CSV files or time-series databases. If auxiliary data like technical indicators and sentiment scores are incorporated, they are collected from relevant sources as well.

2. Data Cleaning: Raw financial data often contains missing values, outliers, and erroneous data points. These need to be addressed to ensure data quality. Missing values might be imputed using interpolation techniques or other appropriate methods. Outliers could be identified using statistical methods and dealt with accordingly, depending on their impact on the analysis.

3. Feature Engineering: In addition to raw stock price and volume data, relevant features are engineered to enrich the dataset. Technical indicators like moving averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD) can provide valuable insights into market trends. Sentiment analysis scores derived from news articles or social media posts can capture market sentiment

LSTM Architecture:



LSTM Architecture

Xt: input time step,

ht: output,

Ct: cell state,

ft: forget gate,

it: input gate,

Ot: output gate,

Äˆt: internal cell state.

Operations inside the light red circle are pointwise

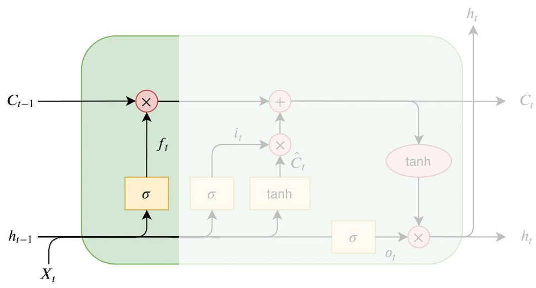
In LSTM architecture, the outcome of an LSTM at a specific moment in time is influenced by three factors:

1. the cell state, which represents the current long-term memory of the network,
2. the previous hidden state, which refers to the output from the prior time step,
3. and the input data present at the current time step.

Long Short-Term Memory neural networks utilize a series of gates to regulate information flow in a data sequence. The forget, input, and output gates serve as filters and function as separate neural networks within the LSTM network. They govern the process of how information is brought into the network, stored, and eventually released.

Forget Gate

The first stage in architecture is Forget Gate. In this stage, the LSTM neural network will determine which elements of the cell state (long-term memory) are relevant based on the previous hidden state and the new input data.



Forget Gate

The previous hidden state (*ht-1)* and the new input data (*Xt)* are input into a neural network that outputs a vector where each element is a value between 0 and 1, achieved through the use of a sigmoid activation function.

This network within the forget gate is trained to produce a value close to 0 for information that is deemed irrelevant and close to 1 for relevant information. The elements of this vector can be thought of as filters that allow more information as the value gets closer to 1.

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

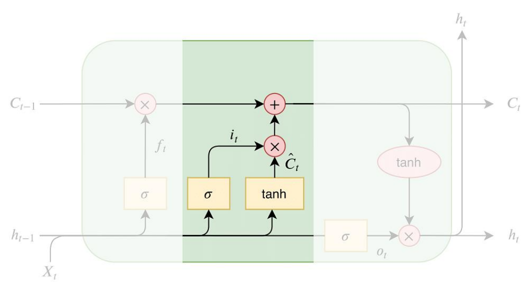
Equation – LSTM Forget Gate

These output values are then multiplied element-wise with the previous cell state (*Ct-1)*. This results in the irrelevant parts of the cell state being down-weighted by a factor close to 0, reducing their influence on subsequent steps.

In essence, the forget gate determines which parts of the long-term memory should be forgotten, given the previous hidden state and the new input data in the sequence.

Input Gate

The following stage involves the input gate and the new memory network. The objective of this step is to identify what new information should be incorporated into the network's long-term memory (cell state), based on the previous hidden state and the current input data.



Input Gate

Both the input gate and the new memory network are individual neural networks in themselves that receive the same inputs, namely the previous hidden state and the current input data. It's important to note that these inputs are the same inputs that are provided to the forget gate.

The input gate is a neural network that uses the sigmoid activation function and serves as a filter to identify the valuable components of the new memory vector. It outputs a vector of values in the range [0,1] as a result of the sigmoid activation, enabling it to function as a filter through pointwise multiplication. Similar to the forget gate, a low output value from the input gate means that the corresponding element of the cell state should not be updated.

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

Equation – LSTM Input Gate

To use stacked LSTM for stock market prediction, you would first need to collect historical data on the stock price. This data would then be used to train the stacked LSTM model. The model would learn to identify patterns in the historical data and use these patterns to predict future stock prices.

The accuracy of stacked LSTM for stock market prediction depends on a number of factors, including the quality of the training data, the complexity of the model, and the market conditions. In general, stacked LSTM has been shown to be more accurate than other methods for stock market prediction, such as ARIMA and ARMA.

Here are some of the steps involved in using stacked LSTM for stock market prediction:

1. Collect historical data on the stock price. This data can be found from a variety of sources, such as Yahoo Finance or Quandl.
2. Prepare the data for training the stacked LSTM model. This may involve cleaning the data, removing outliers, and normalizing the data.
3. Choose the hyperparameters of the stacked LSTM model. These hyperparameters control the complexity of the model and the learning rate.
4. Train the stacked LSTM model on the historical data. This may take several hours or even days, depending on the complexity of the model and the amount of data.
5. Evaluate the performance of the stacked LSTM model on a test set. This will help you to determine how well the model is able to generalize to new data.
6. Use the stacked LSTM model to make predictions.

It is important to note that stacked LSTM is not a perfect predictor of stock prices. The market is constantly changing and there are many factors that can influence stock prices, such as economic news, political events, and natural disasters. As a result, it is important to use stacked LSTM in conjunction with other methods of analysis, such as fundamental analysis and technical analysis.

Here are some of the advantages of using stacked LSTM for stock market prediction:

- It can learn long-term dependencies.
- It can be used for multiple levels of abstraction.
- It can be stacked to form deep neural networks that can learn more complex patterns.

Here are some of the disadvantages of using stacked LSTM for stock market prediction:

- It can be computationally expensive to train.
- It can be sensitive to the choice of hyperparameters.
- It can be difficult to interpret.

Experimental Results

Implementation of LSTM

Training was dome under the following metrics and functions:

- Number of layers =4
- Loss function = Mean Square Error
- Epochs=100
- Batch Size=64

```
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

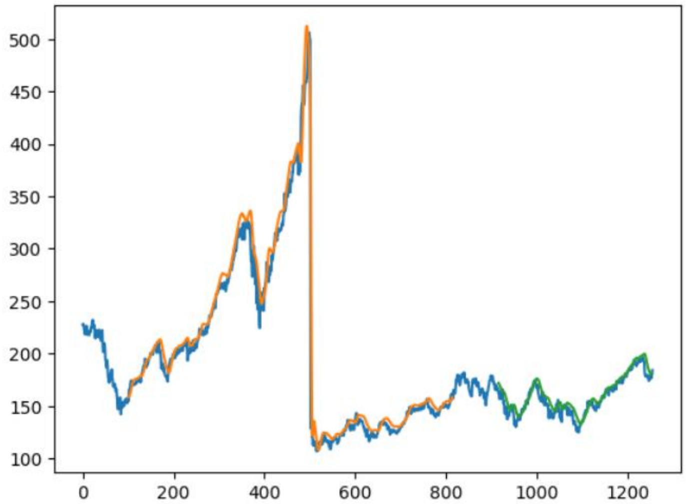
Model Summary

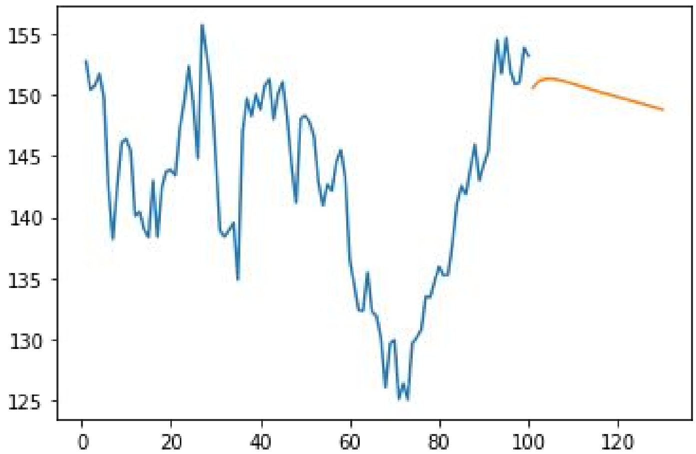
```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
```

Prediction

```
look_back=100
trainPredictPlot = numpy.empty_like(df1)
trainPredictPlot[:,:]=np.nan
trainPredictPlot[look_back:len(train_predict)+look_back,:]=train_predict
testPredictPlot=numpy.empty_like(df1)
testPredictPlot[:, :]= numpy.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(df1)-1,:]=test_predict
plt.plot(scaler.inverse_transform(df1))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```

Output Results





Using LSTM-based stock prediction models for investment decisions on time series data has garnered significant attention due to its potential to capture complex patterns in financial markets. However, it's important to understand both the implications of the findings and the limitations of such models before relying solely on them for investment choices

Implications of Findings:

1. Pattern Recognition: LSTM models have shown the ability to capture intricate temporal patterns in stock price movements. They can potentially identify trends, seasonality, and other recurring behaviors that might be challenging for traditional statistical models to detect.

2. Long-Term Dependencies: LSTMs are designed to handle long-range dependencies in sequences, making them suitable for capturing macroeconomic factors and events that can influence stock prices over time.

3. Feature Learning: LSTM models can automatically learn relevant features from the data, reducing the need for extensive feature engineering. This is particularly useful when dealing with a large number of potential input variables.

4. Adaptability: LSTMs can adapt to changing market conditions and incorporate new information as it becomes available, making them appealing for real-time predictions.

Potential Applications in investment Decisions:

1. Portfolio Optimization: LSTM predictions can be integrated into portfolio optimization strategies to guide the allocation of assets based on predicted future returns and risks.

2. Risk Management: LSTMs can assist in identifying potential market downturns or shifts, allowing investors to adjust their positions and mitigate risks.

3. Algorithmic Trading: Predictions from LSTM models can serve as signals for algorithmic trading strategies, automating the execution of buy and sell orders.

4. Scenario Analysis: Investors can use LSTM-based predictions to run scenario analyses that consider various market conditions and assess their potential impact on investment portfolios.

5. Quantitative Analysis: LSTMs can provide quantitative insights into market behavior, helping analysts understand historical trends and identify anomalies

Limitations and Considerations:

1. Noisy Data: Financial markets are influenced by a multitude of factors, many of which are noisy and unpredictable. LSTMs, like any model, can struggle when confronted with excessive noise.

2. Non-Stationarity: Stock prices often exhibit non-stationary behavior, meaning statistical properties change over time. LSTMs might require careful preprocessing to address this issue.

3. Overfitting: LSTMs can be prone to overfitting, especially when trained on limited data. Regularization techniques are essential to mitigate this risk.

4. Black Box Nature: LSTMs, like other deep learning models, are considered black boxes, making it challenging to interpret their decisions. Understanding the rationale behind predictions is crucial, especially in financial decision-making.

5. Market Efficiency: Financial markets are generally efficient, meaning that publicly available information is quickly reflected in stock prices. Predictive models, including LSTMs, might find it difficult to consistently outperform the market due to this efficiency. 6. Model Robustness: LSTMs' performance can degrade if market conditions significantly deviate from the conditions seen during training.

Conclusion:

In summary, LSTM-based stock prediction models can offer valuable insights for investment decisions, particularly when combined with traditional financial analysis. However, they should not be seen as standalone decision-making tools. Prudent investors should consider their predictions alongside fundamental and macroeconomic analyses, market sentiment, and domain expertise to make informed investment choices.

References

[1] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669.

[2] Brownlee, J. (2018). How to Develop LSTM Models for Time Series Forecasting. *Machine Learning Mastery*.

[3] Akhbari, M., & Banzhaf, W. (2019). Stock price prediction by deep neural networks on single-stock data. *Expert Systems with Applications*, 123, 135-143.

[4] Singhal, M., & Nayak, R. (2020). Predicting Stock Market Trends using Long Short Term Memory Networks. *International Journal of Advanced Computer Science and Applications*, 11(8), 194-200.

[5] Zheng, Z., Zhang, M., & Wang, J. (2017). Stock market trend prediction using a three-dimensional LSTM network. In *International Joint Conference on Neural Networks (IJCNN)* (pp. 1419-1426).

[6] Ren, H., & Yu, X. (2020). Stock Price Prediction Using Time Series LSTM. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5457-5464).

[7] Shi, Z., & Liu, Y. (2019). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *International Conference on Wireless Algorithms, Systems, and Applications* (pp. 267-276).

[8] Zhang, Y., Li, D., & Zhang, J. (2017). Time series prediction for high-frequency stock market data using RNN-LSTM with multiple structures. In *International Conference on Computational Science* (pp. 165-176).

[9] Srinu Vasarao, P., Chakkaravarthy, M. (2022). Time Series Analysis Using Random Forest for Predicting Stock Variances Efficiency. In: Reddy, V.S., Prasad, V.K., Mallikarjuna Rao, D.N., Satapathy, S.C. (eds) Intelligent Systems and Sustainable Computing. Smart Innovation, Systems and Technologies, vol 289. Springer, Singapore. https://doi.org/10.1007/978-981-19-0011-26