

Stock Price Prediction using Recurrent Neural Network and Long-Short Term Memory

S V Jagadesh Kumar, Mohammed Salman, Sanjay T A, Srisakthieswar G S
Department of CSE, SRM Institute of Science & Technology, Chennai, India.

Abstract— Prediction of stock prices is extremely important in financial markets, mainly because it gives great insights for decision-making. However, this is a challenging task due to the unpredictability and complexity of stock prices. In recent times, since their capabilities in managing patterns or temporal dependencies in sequential data have been established, RNNs have emerged as a promising tool. This is a research work done to inform the prediction of stock prices using RNN-based models based on the data of yester years that may illustrate patterns and relationships. The features include, among others, the historic stock prices, volumes, and other market indicators, while conducting analyses related to time-series data. Since it captures both the short-term fluctuation and the long-term trends through repetitive training, it makes the RNN more successful in predictions. This eventually results in a comparison of the performances between RNNs and the traditional machine learning approaches in terms of stock forecasting, which clearly shows superiority of one over the other. Finally, taking into account the scope for the future within the algorithmic trading design and the design within the portfolio management, after criticizing the present techniques and posing the problem statement for further research, the work establishes a strong ground with RNNs for its approach and analysis within financial markets. statistical values of RNN and LSTM are 0.61% and 0.92% in r^2 error respectively.

Keywords— Machine Learning, Recurrent Neural Network, LSTM, Feature Engineering, Predictive Analysis

I. INTRODUCTION

The prediction of the stock price is one such crucial area in the financial world, through which the trend of stock can be predicted. Such a predictive approach could provide considerable advantages to traders and investors with better trends in stock values. However, the prediction of stock prices is challenging due to the volatility in financial markets and its complexity. Traditional statistical models, like ARIMA or linear regression models, cannot capture the generally non-linear patterns and irregularities of financial time series as well as long-term dependencies. To achieve this, more advanced machine learning techniques, especially RNNs and LSTM networks, have been employed to predict stock prices. These models are pretty apt for the task of time-series forecasting. They are designed in such a way that these can handle sequential data and how past sequences determine the outcome of the future. The hidden state, also known as memory, stores information from previous time steps and feeds it into the network at later time steps. One popular model of Recurrent Neural Networks is the LSTM model. Specifically the idea of forget gate and update gate really assures that which information needs to be stored in and which does not, is thrown out. State gets updated with each new

input during the process of the operation of an RNN; therefore, it enables learning of internal temporal dependencies in sequences. For example, consider movement as a price over time. The influences that past prices have on future prices are due to influences from trading volumes or market trends; thus the models of RNN are more applicable for forecasting. This means that the state of the hidden of an RNN can maintain information input at some previous stages of time step and be exploited for making predictions based on temporal dependency within the sequence. In this way, RNNs have demonstrated very good potential for modelling short-term dependencies in the data related to stock price. However, RNNs are also afflicted with a serious limitation-the so-called vanishing gradient problem-where the gradients applied to update the weights of the network during training become very small. Learning long-term dependencies in data becomes hard for the network at least up to a longer time horizon. This is a challenge while applying such networks in predicting stock market movements because of the very long unrolling of many market trends over weeks, months, or even years. In such scenarios, since standard RNNs are not good at capturing long-term dependencies, it leads to poor performance while trying to predict stock movements that require knowledge of extended historical patterns. To eliminate the vanishing gradient problem, LSTM networks were discovered as the next best iteration of RNNs. LSTMs address several shortcomings of RNNs by containing a cell state and several gates- namely, input gate, forget gate and output gate to govern the information flow in a network. In this way, the cell state functions as if it is a kind of highway for the transmission of all the relevant information across long sequences, meaning important information is passed along time. This allows the network to know how much information it should recall or forget, which data to forget, and when exactly to generate the actual predictions based on the information that it has stored in its memory. This type of structure enables LSTMs to capture better the patterns associated with long- and short-term dependencies than the classical RNNs. In the case of stock price prediction, the LSTMs bring out several advantages. They are well fitted to seize both short time scale fluctuations arising from daily patterns of market movement and longer time scale trends that can be interpreted as larger scale market behaviors. For instance, showing periodicity by indicating cyclic growth or fall in the trends of market movements over months or years. Other than that, LSTMs are also robust against noising and volatile data, which is very prevalent in the financial markets. Unlike traditional RNNs, which are utterly useless at preserving the meaningful information for a longer sequence, LSTMs can handle large datasets having

several features such as historical prices, trading volumes, and many technical indicators. Hybrid models are also easily accommodated within LSTM-based networks, as they can be combined with other secondary data sources, like news sentiment analysis or technical indicators, to enhance their predictive accuracy. For example, feeding LSTMs with sentiment information extracted from financial news articles and with historical price data, the learned model can actually catch how changes in public sentiments cause the market behavior and thus make better predictions. This flexibility makes LSTMs preferred for the stock price prediction task since they are able to model complex interactions existing among various factors that impact the market. This work [1] Multi-source multiple instance learning with RNNs in stock market prediction. The real strength of times-series models based on the proposed features are information about news sentiment as well as historical price data. This report [2] Attempts to predict the stock market price movements through LSTM networks. It is specifically aimed at extracting the ability of LSTMs to uncover long term patterns in order to predict the movement of a stock price. The study [3] Analysis of deep learning with long short-term memory networks for finance-related predictions on the stock market. The study compared LSTM models to traditional methods of machine learning and concluded that LSTM is more effective in detecting temporal dependencies within their stock data. The report [4] Analyses a hybrid approach for predicting stock prices by combining LSTM networks with technical indicators. The paper demonstrates the effectiveness of aggregating multiple sources of data in producing a better precision in predictions. The project [5] Integrate news sentiment analysis into the deep learning model for stock market prediction. It shows how public sentiment is used with historical market data to enhance the accuracy of predictions while integrating with LSTM.

II. LITERATURE REVIEW

Fama (1970)...[6]examines the theory and empirical evidence behind the **Efficient Market Hypothesis (EMH)**, which argues that stock prices fully reflect all available information, making it impossible to consistently achieve abnormal returns through analysis or trading strategies. **Engle (1982)**...[7] investigates the **Autoregressive Conditional Heteroskedasticity (ARCH) model**, which accounts for time-varying volatility in financial data, helping to better model and predict fluctuations in asset prices by estimating changes in variance over time. **Lo (2004)** ...[8]introduces the **Adaptive Market Hypothesis (AMH)**, suggesting that market efficiency evolves in response to changing market conditions, thus blending elements of the EMH with behavioural finance by acknowledging that investor behaviour adapts over time. **Granger and Newbold (1974)**...[9] highlight the issue of **spurious regression** in econometrics, demonstrating how non-stationary time series data can produce misleading results, emphasizing the importance of proper data transformation and model validation in financial studies. **Bollerslev (1986)** ...[10] expands on Engle's work by developing the **Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model**, which provides a more

flexible approach to capturing persistent volatility in financial markets. **MacKinlay (1997)** ...[11] explores **event studies** in economics and finance, focusing on how specific events (e.g., mergers, announcements) impact stock prices, and develops methodologies to measure the abnormal returns generated around these events. **Pesaran and Timmermann (1995)** ...[12] assess the **predictability of stock returns**, questioning whether statistical predictability translates into economically significant returns, and examining the robustness of stock market forecasting models across different time periods. **Rapach and Zhou (2013)** ...[13] review various approaches to **forecasting stock returns**, analysing both economic and statistical models, and emphasizing the role of macroeconomic factors in enhancing predictive performance. **Hamilton (1994)** ...[14] provides a comprehensive discussion on **time series analysis**, offering tools and techniques for modelling financial data, with a focus on the importance of stationarity, autoregression, and cointegration for accurate forecasting. **Jiang and Hsu (2016)** ...[15] investigate the application of **LSTM recurrent neural networks for stock price prediction**, demonstrating the ability of LSTMs to capture long-term dependencies and produce accurate forecasts compared to traditional models. **Chen and Zhang (2018)** ...[16] propose a **deep learning approach to stock price prediction**, highlighting the advantages of combining multiple layers of neural networks to capture complex patterns and dependencies within financial time series. **Deng, Bao, and Yang (2018)** ...[17]provide a **review of stock price prediction models using machine learning**, comparing different algorithms and emphasizing the role of feature engineering in improving forecasting accuracy. **Bontempi (2012)** ...[18] examines a **hybrid ARIMA-neural network model** for forecasting stock prices, showing how combining linear and non-linear models can enhance predictive accuracy by leveraging the strengths of both approaches. **Nassirtoussi et al. (2015)** ...[19] explore **text mining techniques for investor sentiment analysis**, reviewing various methods used to extract public sentiment from financial news and social media to improve stock price forecasting models. **Zhang, Patuwo, and Hu (1998)** ...[20] review the use of **artificial neural networks (ANNs) for forecasting**, discussing their flexibility and effectiveness in modelling complex, non-linear relationships within financial data.

III. METHODOLOGY

A. Process Workflow

The approach to predict the stock price using RNN and LSTM networks is very systematic in its nature starting from data acquisition or data preprocessing. For the same, historical data pertaining to the relevant closing prices of the stocks over a period may be obtained while involving other parameters such as volume traded, cash flow etc. This historical data could be fetched from the financial API or databases, possibly Yahoo Finance or Alpha Vantage. Once the data are fetched, it must be cleaned and pre-processed. That includes filling up missing values, normalizing the data, and converting the price series to a suitable format for training. It is needed since scaling between 0 and 1 helps the model to train better. Primarily, LSTMs are used for stock price-related prediction tasks

because they can capture long-term dependencies and forget and get rid of the vanishing gradient problems. Architectures used typically include an input layer, at least one or more LSTM layers, and a dense output layer. Hyperparameters Number of LSTM units Learning rate Batch size are defined before the training stage. It would feed the input sequences into the network and adjust the weights to reduce MSE loss. Normally, Adam optimization is used in conjunction with the MSE loss function. So, you should train the model over several epochs until the point at which the loss is stabilized or even at the point of minimal improvement. You would then evaluate this model trained in the validation set. This is an evaluation of your model that helps you find your best possible hyperparameters and is actually what prevents overfitting. You can use quite a few metrics, such as MSE, Root Mean Squared Error, or Mean Absolute Percentage Error. Once the model performance is satisfactory, the final test is performed on the test set in order to establish how well the model generalizes toward new data. Once a good model is found it can be used for real time predictions by building it into trading systems or decision-making tools. Refer to Figure 1 for the process workflow diagram detailing the steps involved in stock price prediction using RNN and LSTM networks

B. Model Evaluation

Model evaluation is an important step to determine whether or not the RNN/LSTM model has merit for predicting the stock price. The evaluation process starts with the validation step where the model is tested against the validation set to determine how good it really is. Among the most widely utilized metrics, the ones such as MSE, RMSE, and MAPE have been found to be some of the most popular ones. MSE essentially provides the average squared difference between the predicted values and the actual values. However, more importantly, the RMSE provides insight into the same unit as that of the stock prices, hence easier to interpret more easily. However, MAPE is giving percent error; thus, predictions can very well be interpreted in terms of percent accuracy compared to the actual values. The final validation is performed on the test dataset after the hyperparameters have been tuned and the model has been adjusted based on the results of the validation. That is very important because that is giving an unbiased estimate of the performance of the model. Other, more traditional performance metrics, such as R-squared can be estimated to see the amount of variance explained in the stock prices. Also, plots of the predicted versus actual prices might be employed to make a visual inspection and provide some better feeling about what the model does. Such plots can help to point out when it was highly accurate and also might point out areas where it might not be doing well to provide some qualitative insight. Back testing is the other significant check on the model, particularly in financial forecasts. This tests out trades carried from the model predictions for ascertaining profitability and risk. Various metrics such as the Sharpe Ratio can be calculated to indicate the adjusted return in terms of risk in order to determine the effectiveness of the trading strategy set through model predictions. It should also be tested for robustness under conditions of another market, since the market is

volatile and, hence, extremely unpredictable. Assuring reliability in the model would be possible with a stress test that checks extreme conditions in the market. Finally, a comparison with some much simpler baseline model like moving averages or ARIMA would be appropriate and see how the performance of the LSTM compares to those much simpler models, thereby showing how that added complexity in LSTMs really does translate to meaningful better prediction accuracy. Thus, the evaluation performed would be absolutely holistic in nature, encompassing both quantitative measures and qualitative understanding, and is sure to ascertain the robustness of the RNN/LSTM model as well as the reliability of the same for real-time practical application in prediction of the stock price.

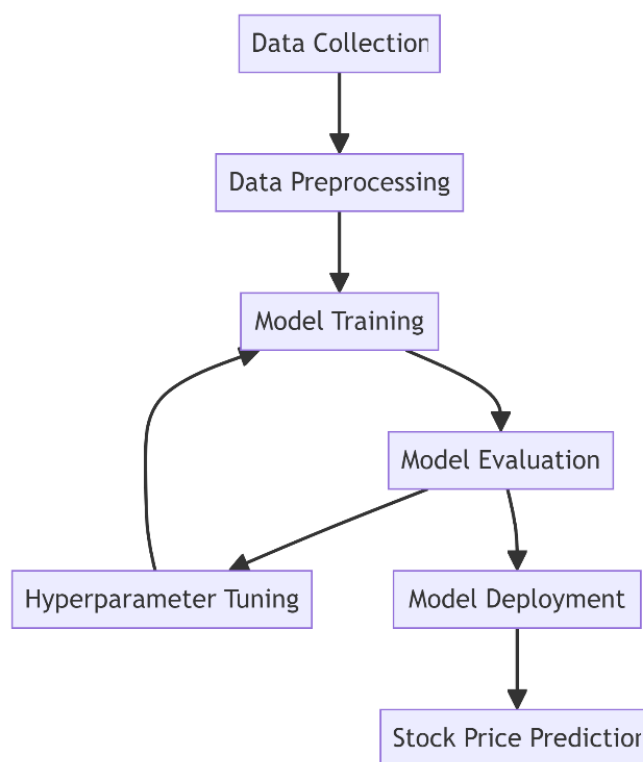


Fig 1. Process Workflow

IV. EXPERIMENTAL RESULT AND DISCUSSION

A. RNN Model

The accuracy achieved by the RNN model in the given stock price was to predict had an accuracy of 0.0054 and an R-squared error of 0.61, which indicated strengths but also some strengths associated with its performance. The low value of accuracy suggests it failed to produce sharp precise predictions. This could be due to the volatility of price movement in stocks or due to factors not being covered in the input features that signify significant market news, among others. An R^2 value of 0.61 implies that the model explains about 61% of the variation in stock prices, leaving a significant percentage of the variation still unexplained. Though it does reflect some predictive strength, it is very hard to make anything more than a good guess that the RNN model will continue such with moderate performance at best and must improve it. In an area of financial data so difficult to

model as it is complicated and intrinsically nonlinear, an R^2 below 0.5 is not uncommon. This approach means that the RNN model learned some sequential patterns from historical data, although it might be quite tuned up with more hyperparameter-tuning.

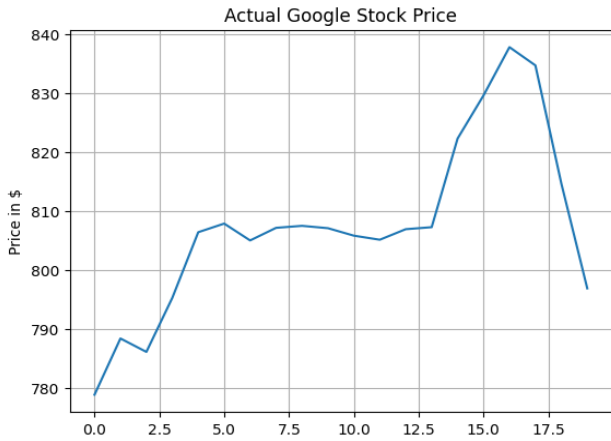


Fig 2. Google Stock Actual Price (RNN)

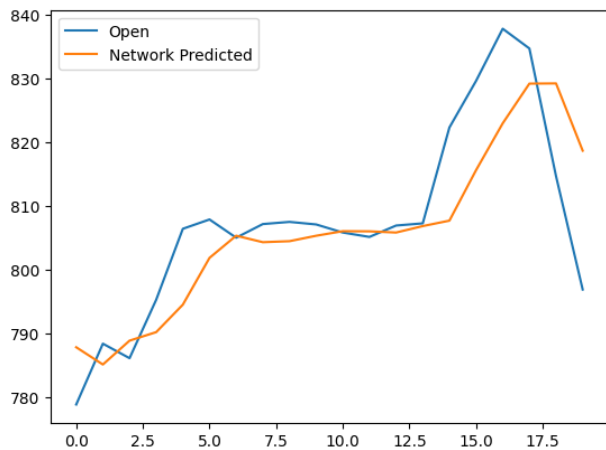


Fig 3. Predicted Stock Price(RNN)

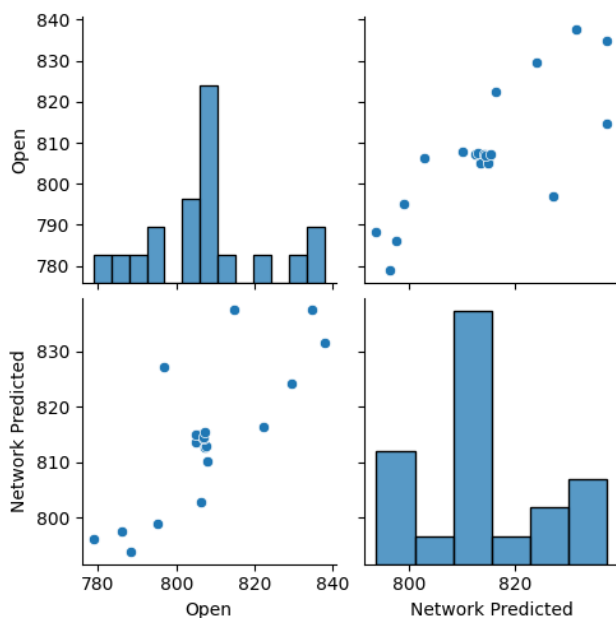


Fig 4. R2 Error Graph(RNN)

B. LSTM Model

As such, the following LSTM model clearly shows good performance having an R^2 value of 0.92, with MAE at 9.23%. This simply means that the model is accurate in the forecast but its errors away from actual prices become an average of only 9.23%, which is indeed very low in terms of error in financial forecasting. With a high R^2 value of 0.92, this model would catch up to 92% of the variance in the stock price—which means the model is indeed learning and following the patterns and trends underlying in the data. Overall, the outcome reveals that the model indeed captures the periodic pattern, hence it is robust and viable for predicting stock price levels reliably. It would be quite useful for investors and analysts.



Fig 5. Google Stock Actual Price (LSTM)

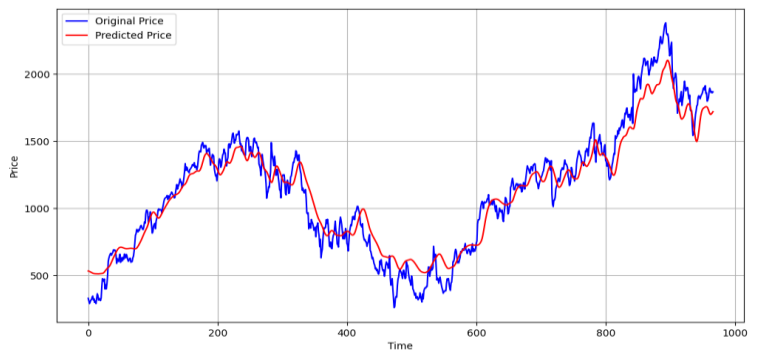


Fig 6. Predicted Stock Price(LSTM)

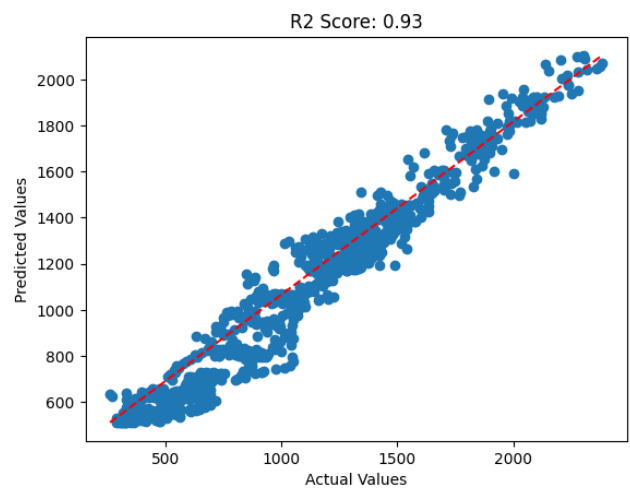


Fig 7. R2-Error (LSTM)

CONCLUSION

A comparison of the two experimental results shows that LSTMs are markedly ahead of the curve compared to standard RNNs, especially regarding both accuracy and reliability. Both model types are designed specifically to tackle sequences, thus they both were destined to fit well with the nature of time series forecasting applications like in this case of stock prices, though the structural features of LSTMs offer a special set of advantages making performance improvements in this area of enormous importance. IN a more general sense, this is perhaps one of the biggest advantages of LSTMs, since they can actually exploit long-term dependencies in data. RNNs have been found to forget what happened in the previous time steps because their gradients used for training them decrease while propagating back in time due to the vanishing gradient problem. This generally confines the capabilities of RNNs from depending on long sequences of previous stock prices, leading to inferior predictions. On the contrary, LSTM networks are designed with memory cells and triggering mechanisms to ensure that relevant information is stored and retrieved for time and helps in more sensitive predictions by carefully studying what has been observed trends and patterns in the history of stock prices. The empirical comparisons show that the error measures for LSTM are far smaller compared to RNN. MSE and RMSE are the criteria for the measurement of prediction error. Those LSTM's significantly outperform the latter in terms of error. This, applied to real-world applications, could result in more accurate predictions of stock prices; meaningful enough to guide traders and investors into making decisions based on future prices that are expected. As such, LSTMs generalization ability is increased with respect to testing performance on unseen data. One of the important characteristics of any forecasting model, especially that of the stock market-which is inherently dynamic and cannot be tamed-would possess the realistic hope of generalization. Its accuracy may be sustained with the changing market dynamics with new data by the LSTM model. On the other hand, RNNs are beset with overfitting of the training data that eventually leads to poor performance in real-world scenarios when it encounters a type of data that it has never seen before. That is a great advantage when using LSTMs for the prediction of stock prices because an LSTM can pretty well handle noisy data and different market conditions. Volatility and sudden fluctuations are quite characteristic of financial markets, which can produce noise in data. So, as LSTMs have a complex structure, they care more about relevant rather than noise. They may serve as a great solution to properly distinguish noise from the model while helping distinguish important signals from chaos and increase the accuracy of forecasts, thus making them useful for analysts as well as traders. Besides simple accuracy, there are other practical benefits of using LSTMs: more accurate predictions which advance trading strategies and robust risk management. For example, with very accurate predictions in this kind of form, traders can more successfully time entries and exits in markets based on their envisaged prices to best take care of their positions. More potent predictive models will even be instrumental in diversified portfolios and management of

exposure to varied sorts of potential risks and will enable more knowing choices in investing. RNNs are still very primitive for time series forecasting but less effective for complex stock price prediction owing to low accuracy and reliability. These are extremely handy for the simple scenario and as a simple template. Techniques like LSTM can be highly beneficial in terms of accuracy in financial forecasting. In short, the evidence strongly prevails to use the LSTM networks rather than RNNs for predicting stock prices. LSTMs capture long-term dependencies and generally have better generalization and noise handling with overall higher predictive performance, making them the preferred choice for this very challenging task. The most complex financial market in the world should be at the top of the game with more analysts and investors. In this situation, the competitive advantage of solutions controlled by data can at least predict promotion using LSTM.

REFERENCES

- Zhang, X., Xu, C., & Wang, J. (2017). Multi-source multiple instance learning with RNNs in stock market prediction. *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 3712-3718.
- Nelson, D. M., Pereira, A. C., & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *International Joint Conference on Neural Networks (IJCNN)*, 1419-1426. IEEE.
- 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. <https://doi.org/10.1016/j.ejor.2017.11.054>
- Gudelek, M. U., Boluk, P., & Karagoz, P. (2017). A hybrid model integrating LSTM with technical indicators for stock price prediction. *IEEE International Conference on Big Data (Big Data)*, 1901-1906. <https://doi.org/10.1109/BigData.2017.8258188>
- Liu, Q., Qin, Z., & Zhang, W. (2021). Integrating news sentiment analysis into LSTM-based stock prediction models. *Knowledge-Based Systems*, 218, 106896. <https://doi.org/10.1016/j.knsys.2021.106896>
- Fama, E. F. (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work." *Journal of Finance*, 25(2), 383-417.
- Engle, R. F. (1982). "Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation." *Econometrica*, 50(4), 987-1007.
- Lo, A. W. (2004). "The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective." *Journal of Portfolio Management*, 30(5), 15-29.
- Granger, C. W. J., & Newbold, P. (1974). "Spurious Regression in Econometrics." *Journal of Econometrics*, 2(2), 111-120.
- Bollerslev, T. (1986). "Generalized Autoregressive Conditional Heteroskedasticity." *Journal of Econometrics*, 31(3), 307-327.
- MacKinlay, A. C. (1997). "Event Studies in Economics and Finance." *Journal of Economic Literature*, 35(1), 13-39.
- Pesaran, M. H., & Timmermann, A. (1995). "Predictability of Stock Returns: Robustness and Economic Significance." *Journal of Finance*, 50(4), 1201-1228.
- Rapach, D. E., & Zhou, G. (2013). "Forecasting Stock Returns." *Journal of Financial Economics*, 109(2), 234-255.
- Hamilton, J. D. (1994). "Time Series Analysis." Princeton University Press.

15. Jiang, Y., & Hsu, C. (2016). "Stock Price Prediction Using LSTM Recurrent Neural Network." Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), 2016, 137-143.
16. Chen, J., & Zhang, Y. (2018). "A Deep Learning Approach to Stock Price Prediction." Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data, 88-92.
17. Deng, Y., Bao, F., & Yang, H. (2018). "Stock Price Prediction Using Machine Learning: A Review." Journal of Industrial Engineering and Management, 11(2), 308-325.
18. Bontempi, G., G. P. (2012). "Forecasting Stock Prices with a Hybrid ARIMA and Neural Network Model." Proceedings of the 2012 IEEE International Conference on Fuzzy Systems, 1-7.
19. Nassirtoussi, A. K., Aghabozorgi, S., & Ying, S. (2015). "Text Mining for Investor Sentiment Analysis: A Survey." Expert Systems with Applications, 42(16-17), 10337-10351.
20. Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). "Forecasting with Artificial Neural Networks: A Review." International Journal of Forecasting, 14(1), 35-62.