Smart Stock Price Prediction Algorithm using RNN variant Long Short-Term Memory

Hemank Jain*

SCSE, Galgotias University, Greater Noida, 203201, Uttar Pradesh, India

* Corresponding author

doi: https://doi.org/10.21467/proceedings.178.1

ABSTRACT

Stock price prediction plays a critical role in helping individuals and organizations make informed financial decisions. This research introduces an innovative model based on Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN), designed to forecast stock prices. The model leverages historical stock data, including key indicators such as opening and closing prices, daily highs and lows, and trading volumes. To ensure reliable predictions, the study incorporates a comprehensive preprocessing pipeline. This pipeline handles data cleaning and normalization to prepare the input data for analysis. The core of the model is built on advanced RNN architectures like LSTM and Gated Recurrent Units (GRUs), which are well-suited for capturing complex temporal patterns in sequential data—an essential aspect of accurate stock price forecasting. The model's performance is evaluated using standard error metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). These metrics provide a clear measure of the model's accuracy and reliability. The study highlights the power of RNNs in stock price prediction and introduces an interactive, user-friendly tool tailored for investors and traders, enabling them to refine their financial strategies and make better decisions.

Keywords: Prediction, Validation, Precision, RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory)

1 Introduction

Predicting the stock market is a notoriously complex task because countless factors influence its movements. Core elements like supply and demand for stocks, broader market trends, the overall health of the economy, and the performance of individual companies all play significant roles. However, these factors do not operate in isolation they interact in dynamic and often unpredictable ways. Adding to the complexity is the emotional and psychological aspect of investing. Investor sentiment, or how people feel about certain stocks or the market, can cause sudden shifts in stock prices. For instance, optimism about a company's future can drive its stock higher, while fear or uncertainty can trigger sell-offs. Additionally, major news events, whether political, economic, or related to a specific industry, can disrupt the market at any moment, creating volatility that is difficult to foresee. To navigate this uncertainty, investors use a variety of tools and strategies to try to predict future market movements. Fundamental analysis, for example, focuses on studying a company's financial health, such as its earnings, revenue, and growth potential, to determine whether its stock is a good investment. On the other hand, technical analysis looks at historical price patterns and trends to anticipate future behaviour. While these methods provide valuable insights, they are far from foolproof. The stock market is influenced by countless variables, and even the most sophisticated tools and strategies cannot account for every possible scenario. Unexpected events, like the introduction of groundbreaking technology, sudden regulatory changes, or global crises, can dramatically shift market

Proceedings DOI: 10.21467/proceedings.178; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-984081-8-1

dynamics in ways no one could have anticipated. Given this inherent unpredictability, it is essential for investors to adopt strategies that help mitigate risk. Diversification, or spreading investments across different asset classes, sectors, and regions, is one of the most effective ways to reduce exposure to any single market event. Risk management also involves setting realistic goals, maintaining discipline during market fluctuations, and avoiding emotional decision-making. Perhaps most importantly, a long-term investment approach often proves beneficial. While short-term market movements can be volatile and stressful, history has shown that markets tend to trend upward over time. By focusing on the bigger picture and staying committed to a well-thought-out investment plan, investors can navigate uncertainty and position themselves for success in the long run.

Addressing Specific Questions like Predicting Day-Ahead Stock Prices, Forecasting stock prices for the following day based solely on historical price data is a challenging task due to the complex and dynamic nature of financial markets. However, advancements in machine learning, particularly Recurrent Neural Networks (RNNs) equipped with Long Short-Term Memory (LSTM) architecture, offer a promising solution. These models are designed to analyse historical price patterns and identify underlying trends, allowing them to make predictions about future price movements. By capturing the sequential and temporal dependencies within stock market data, RNNs with LSTMs show significant potential in improving the accuracy of day-ahead stock price predictions and the other question is validating the Model's Performance, Validation is a critical step in determining the accuracy and reliability of any predictive model. For this study, the model employs multiple validation techniques to ensure robust performance. One key approach is back testing, where the model is tested on historical data to evaluate how well it predicts past stock prices. In addition, methods such as cross-validation and out-of-sample testing are applied to assess the model's ability to generalize to new, unseen data. These techniques help ensure that the model performs consistently and can provide precise predictions in real-world scenarios, making it a dependable tool for investors and traders. The Role of Machine Learning in Stock Market Forecasting is that the stock market is a highly dynamic and intricate environment where investors are constantly searching for ways to predict future price movements more accurately. Traditional methods, such as fundamental and technical analysis, often fall short due to the market's complexity, volatility, and sensitivity to a wide range of factors. In recent years, artificial intelligence (AI) and machine learning (ML) have emerged as transformative technologies, providing new opportunities to enhance predictive accuracy in stock price forecasting. Among the many ML techniques available, Recurrent Neural Networks (RNNs) have proven particularly effective for time-series analysis. Their ability to capture temporal dependencies within sequential data makes them well-suited for modelling stock market behaviour. LSTM, a specialized RNN architecture, addresses common challenges such as short-term memory loss in standard RNNs, making it ideal for analysing stock price movements over time. These capabilities enable RNNs to uncover intricate patterns and trends in stock market data, which are often missed by traditional methods.

This paper focuses on harnessing the power of RNN-based algorithms, particularly LSTMs, to develop a robust and reliable stock price prediction model named **"ProfitPlus."** By utilizing historical stock market data, the model is designed to predict future price movements with improved accuracy. ProfitPlus aims to address the limitations of traditional forecasting methods, offering a modern, AI-driven approach to navigating the complexities of financial markets. Through rigorous validation and testing, this paper demonstrates the model's effectiveness, showcasing its potential as a valuable tool for investors and traders seeking to make more informed financial decisions. Some of the key objectives of this study includes, Model Development, the study focuses on designing and implementing an RNN-based algorithm specifically tailored for predicting stock prices of Google and Tesla. ProfitPlus leverages historical stock market data, including key indicators such as price trends, trading volumes, and other relevant metrics. By analysing these inputs, ProfitPlus is trained to identify patterns and optimize its predictive capabilities, offering accurate and reliable forecasts for these high-profile stocks. Other objective of the study is Feature Engineering, to enhance the performance of the ProfitPlus model, this study applied advanced feature engineering techniques to derive meaningful insights from raw data. This includes integrating technical indicators and incorporating historical data from multiple years to capture long-term trends. These engineered features provide the model with a richer understanding of the stock market's behaviour, ultimately improving its predictive accuracy and robustness. Third objective of the study is Evaluation and Comparison, the effectiveness of ProfitPlus is rigorously evaluated through a comprehensive comparison with existing benchmark methods. These benchmarks include traditional statistical models and other machine learning approaches commonly used for stock price prediction. Through detailed empirical analysis, the study assess the strengths and weaknesses of RNNs in handling stock market data and demonstrate how ProfitPlus performs in comparison to these established methods, Finally the last object is Real-World Applications and Implications, the paper explores the practical implications of the findings and their relevance to real-world investment strategies. By showcasing the effectiveness of RNN-based algorithms in predicting stock prices, this paper aims to provide actionable insights that investors can use to make more informed financial decisions. ProfitPlus highlights the potential of advanced machine learning techniques to improve forecasting accuracy, empowering traders and investors to better navigate the complexities of financial markets.

2 Literature Review

Long Short-Term Memory (LSTM) networks, a type of deep recurrent neural network [1], play a central role in this Paper. Unlike traditional feed-forward networks, recurrent neural networks (RNNs) allow data to flow in multiple directions. This structure enables information to be retained and passed between layers, either to previous layers or within the same layer. This capability allows RNNs, and particularly LSTMs, to maintain both short-term and long-term memory, making them highly effective in learning from sequential data. LSTM networks were originally introduced to address a significant challenge faced by traditional RNNs: the vanishing gradient problem [2]. This issue arises when training networks with long data sequences, where the gradients become too small for effective learning. LSTM networks overcome this limitation using specialized components called "gates." These gates regulate the flow of information, control error propagation, and allow the network to selectively adjust weights. They can even truncate gradients when necessary, ensuring efficient and accurate training. Since their inception, LSTM networks have shown exceptional performance in various applications. They are widely recognized as state-of-the-art models in fields like Natural Language Processing (NLP), excelling in tasks such as handwriting recognition. While several variations of the original LSTM architecture have been proposed over time, comparative studies suggest that most offer only marginal improvements over the original design [3]. In the realm of stock price prediction, LSTM networks have been utilized in a variety of ways. Some approaches focus on leveraging textual data, such as news articles, to forecast stock price trends [4]. Others, like this paper, use price data itself to predict future movements. The adaptability and robustness of LSTM networks make them an invaluable tool for tackling the complexities of financial market forecasting. In the real time of financial forecasting, researchers are increasingly leveraging advanced computational techniques to predict stock price movements and other financial trends. For example, one study uses historical price data and stock market indexes to predict whether a stock price will rise, fall, or remain unchanged on a given day. Another study compares the performance of Long Short-Term Memory (LSTM) networks and Multilayer Perceptron (MLP) models with a novel method that combines wavelets and Convolutional Neural Networks (CNNs). While their proposed method outperforms both LSTM and MLP networks, the results are closely aligned with those achieved by the LSTM model, emphasizing its reliability and effectiveness [5]. These advanced methods, like CNNs and LSTMs, represent significant progress in how computers analyse and interpret complex financial data. CNNs excel at identifying intricate patterns in data, much like finding hidden details in a large, complex image. Meanwhile, LSTMs are particularly adept at working with sequential data, allowing them to understand how financial trends evolve over time. Together, these techniques showcase how modern technology is transforming the ability to analyse financial markets and uncover meaningful insights [6].

A Simplified Comparison, When it comes to analysing complex financial data, Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks each bring unique strengths to the table. CNNs are like expert pattern finders—they excel at spotting critical details within vast amounts of information, much like finding a needle in a haystack. This makes them particularly useful for identifying specific features in data [7]. LSTMs, on the other hand, are memory specialists. They are designed to remember important events from the past and use that information to make accurate predictions about the future. This ability to capture and utilize historical context makes LSTMs especially effective for time-series data, such as stock price movements, where understanding how trends evolve over time is crucial. Applications of LSTMs in Financial Prediction LSTMs have become a popular tool for predicting stock prices and analysing financial trends. They have shown considerable success in helping researchers and analysts make informed forecasts. One notable application is combining LSTM networks with other techniques to improve predictions. For example, these hybrid methods have been used to analyse relationships between different stocks by leveraging historical data, leading to more accurate insights into how stocks influence one another [8]. By using their ability to process sequential data and retain critical information over time, LSTMs have proven to be a powerful resource for navigating the complexities of financial markets and enhancing predictive accuracy. While Long Short-Term Memory (LSTM) networks are powerful tools for analysing sequential data, they are not without their limitations. One common issue is that LSTMs can sometimes struggle to retain all the necessary information needed for accurate predictions. This can result in an incomplete understanding of the broader context, making it harder to identify patterns and trends accurately. Such limitations highlight the need for enhancements to improve the performance of LSTMs, particularly in complex tasks like stock price prediction [9]. Bi-Directional LSTM and CNN-BiLSTM Hybrid Models, to address the above challenges challenges, researchers have developed advanced variations of LSTM, such as Bi-Directional LSTM (BiLSTM). Unlike standard LSTMs, which only process information in one direction (from past to future), BiLSTM networks consider information from both the past and the future. This bidirectional approach provides a more comprehensive understanding of the data, resulting in more accurate predictions [10]. Further advancements combine the strengths of Convolutional Neural Networks (CNNs) with BiLSTM networks. By integrating CNNs—which excel at identifying important details in complex data—with the memory capabilities of BiLSTM, these hybrid models can deliver even better results. The CNN component efficiently extracts critical features, while the BiLSTM layer processes this information in both directions, ensuring that key patterns and contextual dependencies are not missed. Together, these innovations push the boundaries of predictive accuracy, offering powerful tools for tackling the complexities of financial forecasting and other time-series applications [11]. The Superiority of the CNN-BiLSTM Combination Model [12], has demonstrated that combining Convolutional Neural Networks (CNNs) with Bi-Directional Long Short-Term Memory (BiLSTM) networks significantly improves stock price prediction compared to using either method alone. This hybrid approach harnesses the unique strengths of both models: CNNs excel at identifying critical patterns and features within complex data, while BiLSTMs provide a deeper understanding of sequential dependencies by analyzing data from both past and future perspectives. By integrating these advanced techniques, the CNN-BiLSTM model delivers more accurate predictions, making it a valuable tool for financial analysis. This improvement in forecasting capability empowers investors to make smarter decisions, enhancing their ability to navigate the complexities of the stock market and optimize investment strategies.

3 Research Methodology

Cracking the Code of Recurrent Neural Networks (RNNs), RNNs are like intelligent assistants for computers, enabling them to learn from and remember past experiences. Unlike standard computer systems that process data all at once, RNNs work sequentially, processing information piece by piece. This makes them particularly well-suited for tasks that require an understanding of patterns over time, such as predicting the next word in a sentence or forecasting stock market trends using advanced machine learning techniques. Think of an RNN as a collaborative team of specialists. One "team member" is responsible for taking in the data, other processes it, and a final member delivers the outcome. These specialists work together seamlessly, passing information back and forth like a relay team. Each layer of the network contributes its own expertise, allowing RNNs to break down complex problems into smaller, more manageable steps. The true power of RNNs lies in their ability to recognize patterns in data that changes over time. They achieve this by remembering past events and using that knowledge to predict what might happen next. It's like having a crystal ball that uses past experiences to make informed guesses about the future. However, before RNNs can perform their "magic," the data must go through a process called preprocessing. This step involves cleaning, organizing, and structuring the data to make it easier for the RNN to analyse. Think of it as setting the stage for a performance—everything needs to be in the right place for the show to run smoothly. RNNs are transformative because they can learn from the past to make predictions about the future. This capability allows them to tackle a wide range of real-world challenges, from understanding human language to making smarter financial decisions. By processing sequences of data, RNNs help computers approach problems in a way that feels more human, offering powerful solutions for interpreting and analysing information over time. Long Short-Term Memory (LSTM), LSTM networks represent a major advancement in the world of neural networks, improving upon traditional Recurrent Neural Networks (RNNs) by addressing the problem of forgetting important information over long sequences. Traditional RNNs are prone to what is known as the "vanishing gradient problem," where the network struggles to retain crucial details as it processes increasingly long data sequences. LSTM overcomes this limitation by introducing a special "memory" system that allows it to store and recall important information for much longer periods. The way LSTM works is through a three-step process: forgetting, selecting, and using memories. First, it decides which information to discard, much like clearing out unnecessary clutter from a closet. Then, it selects the most important memories to keep, like deciding which items are useful enough to hold on to. Finally, in the output stage, LSTM makes decisions based on those selected memories, almost like picking the right outfit for the day based on the weather and occasion. By managing and organizing information in this way, LSTM ensures that only the most relevant data is used to make predictions or decisions, which makes it highly effective for complex tasks. Its ability to remember important details over long time spans is crucial in fields such as natural language processing, speech recognition, and stock market prediction, where understanding long-term dependencies is key to accuracy (Fig 1). In essence, LSTM functions as a smart assistant that can process and retain critical past information, helping machines learn more effectively and make accurate predictions.

```
#Modeling
model=Sequential()
model.add(LSTM(units=50,
               activation="relu",
               input_shape=(X_train.shape[1],lookback)))
model.add(Dropout(0.2))
model.add(Dense(1))
```

Fig 1 Modeling Structure of ProfitPlus

4 Implementation

ProfitPlus is an innovative platform built to help users analyse stock market trends and make predictions about future prices. It uses advanced machine learning techniques, particularly **Long Short-Term Memory (LSTM)** networks, which are a type of deep learning model well-suited for time-series forecasting like stock prices. The platform is built using **powerful Python libraries**. For data handling and manipulation, it employs **Pandas**, a library known for its ability to efficiently manage large datasets. To build and train the LSTM model, it leverages **Keras**, which is a user-friendly deep learning framework. Specifically, Keras components like **Sequential** (for stacking layers) and **Dense** (for creating fully connected layers) are used in building the model architecture. Together, these technologies enable **ProfitPlus** to make highly accurate predictions. At the same time, the platform ensures that users enjoy an intuitive and efficient experience. Both the **user interface (UI)** and the **user experience (UX)** are carefully designed to make the platform easy to use, regardless of a user's technical expertise. This makes **ProfitPlus** an accessible and powerful tool for anyone looking to analyse the stock market and predict future movements.

The development of the ProfitPlus model follows a clear and structured approach. First, historical stock price data from companies like Google and Tesla is gathered. This includes key details such as the opening price, closing price, highest and lowest prices, and the trading volume (Fig2, Fig3 for a visual representation of this data). Once this data is collected, it is split into two parts: a training set and a test set (Fig4, Fig5). The training set is used to teach the ProfitPlus model how to identify patterns and trends in the stock prices, while the test set is used later to check how well the model learned from the training data. Before the ProfitPlus model can start learning, the data needs to be cleaned and organized—a process known as preprocessing (Fig 6, Fig7). This step is crucial because it ensures that the data is free from errors or irrelevant information, allowing the ProfitPlus to perform more accurately. After the data has been prepped, the next phase is selecting the right machine learning model and building it (Fig8). This is a critical step, as choosing the right model is key to making accurate predictions. Once the ProfitPlus is built, it undergoes a training process, where it learns from the historical data. Afterward, it is evaluated to ensure that it is performing as expected. Testing the ProfitPlus comes next, where it is challenged with new data to see how well it can predict stock prices in real-world scenarios. Finally, the results of the ProfitPlus are displayed in an easy-to-understand visual format (Fig9), which allows users to interpret the data and make informed decisions about stock market investments. This entire process, from data collection to visualization, is designed to help users better understand the trends and movements in the stock market, empowering them to make smarter financial choices.

```
            Date         Open         High          Low        Close  \
0     2013-01-02   357.385559   361.151062   355.959839   359.288177
1     2013-01-03   360.122742   363.600128   358.031342   359.496826
2     2013-01-04   362.313507   368.339294   361.488861   366.600616
3     2013-01-07   365.348755   367.301056   362.929504   365.001007
4     2013-01-08   365.393463   365.771027   359.874359   364.280701
...          ...          ...          ...          ...          ...
1254  2017-12-22  1061.109985  1064.199951  1059.439941  1060.119995
1255  2017-12-26  1058.069946  1060.119995  1050.199951  1056.739990
1256  2017-12-27  1057.390015  1058.369995  1048.050049  1049.369995
1257  2017-12-28  1051.599976  1054.750000  1044.770020  1048.140015
1258  2017-12-29  1046.719971  1049.699951  1044.900024  1046.400024

        Adj Close   Volume
0      359.288177  5115500
1      359.496826  4666500
2      366.600616  5562800
3      365.001007  3332900
4      364.280701  3373900
...           ...      ...
1254  1060.119995   755100
1255  1056.739990   760600
1256  1049.369995  1271900
1257  1048.140015   837100
1258  1046.400024   887500

[1259 rows x 7 columns]
```

Fig 2 Google historical data

```
In [2]: data=pd.read_csv("C:/Users/HP/TSLA.csv")

In [3]: data.head()
Out[3]:
```

	Date	Open	High	Low	Close	Adj Close	Volume
0	2010-06-29	19.000000	25.00	17.540001	23.889999	23.889999	18766300
1	2010-06-30	25.790001	30.42	23.299999	23.830000	23.830000	17187100
2	2010-07-01	25.000000	25.92	20.270000	21.959999	21.959999	8218800
3	2010-07-02	23.000000	23.10	18.709999	19.200001	19.200001	5139800
4	2010-07-06	20.000000	20.00	15.830000	16.110001	16.110001	6866900

Fig 3 Tesla historical data

```
[3]: trainset = df.iloc[:,1:2].values

[4]: trainset

[20]: dataset_test =pd.read_csv("C:/Users/HP/testset.csv")
```

Fig 4 Train set, Test set for Google stocks

```
In [20]: tesla_data[0:5]

Out[20]: array([[23.889999],
                [23.83    ],
                [21.959999],
                [19.200001],
                [16.110001]])

In [21]: tesla_data=tesla_data.astype("float32")

In [22]: def split_data(dataframe,test_size):
             pos=int(round(len(dataframe)*(1-test_size)))
             train=dataframe[:pos]
             test=dataframe[pos:]
             return train,test,pos

In [23]: train,test,pos=split_data(tesla_data,0.20)

In [24]: print(train.shape,test.shape)
         (1933, 1) (483, 1)
```

Fig 5 Train set, Test set for Tesla stock

```
In [20]: dataset_test =pd.read_csv("C:/Users/HP/testset.csv")

In [21]: real_stock_price = dataset_test.iloc[:,1:2].values

In [22]: dataset_total = pd.concat((df['Open'],dataset_test['Open']),axis = 0)
         dataset_total

Out[22]: 0       357.385559
         1       360.122742
         2       362.313507
         3       365.348755
         4       365.393463
                    ...
         120    1143.599976
         121    1128.000000
         122    1121.339966
         123    1102.089966
         124    1120.000000
         Name: Open, Length: 1384, dtype: float64
```

Fig 6 Processing Google Stocks Data

```
In [8]: #preparing the data
        tesla_data=data[["Date","Close"]]

In [9]: tesla_data.head()
```

Out[9]:

	Date	Close
0	2010-06-29	23.889999
1	2010-06-30	23.830000
2	2010-07-01	21.959999
3	2010-07-02	19.200001
4	2010-07-06	16.110001

Fig 7 Processing Tesla Stocks Data

```
In [11]: from keras.models import Sequential
         from keras.layers import Dense
         from keras.layers import LSTM
         from keras.layers import Dropout

In [12]: regressor = Sequential()
         regressor.add(LSTM(units = 50,return_sequences = True,input_shape = (x_train.shape[1],1)))

In [13]: regressor.add(Dropout(0.2))

In [14]: regressor.add(LSTM(units = 50,return_sequences = True))
         regressor.add(Dropout(0.2))

In [15]: regressor.add(LSTM(units = 50,return_sequences = True))
         regressor.add(Dropout(0.2))

In [16]: regressor.add(LSTM(units = 50))
         regressor.add(Dropout(0.2))

In [17]: regressor.add(Dense(units = 1))

In [18]: regressor.compile(optimizer = 'adam',loss = 'mean_squared_error')
```

Fig 8 Implementing LSTM

```
[33]: plt.plot(real_stock_price,color = 'red', label = 'Real Price')
      plt.plot(predicted_price, color = 'blue', label = 'Predicted Price')
      plt.title('Google Stock Price Prediction')
      plt.xlabel('Time')
      plt.ylabel('Google Stock Price')
      plt.legend()
      plt.show()

In [72]: plt.figure(figsize=(14,5))
         plt.plot(result_data,label="Real Values")
         plt.plot(train_prediction_data_copy["Predicted"],color="brown",label="Train Predicted")
         plt.plot(test_prediction_data["Predicted"],color="red",label="Test Predicted")
         plt.xlabel("Time")
         plt.ylabel("Stock Values")
         plt.legend()
         plt.show()
```

Fig 9 Visualizing the estimations

- **Minimum Requirement Specifications**

The implementation of ProfitPlus is done in the system with following specifications and software.

Hardware:

• Processor: Pentium IV 2.4 GHz

• Hard disk: 60 GB

• RAM: 1 GB
• Monitor: 15" Colour
• CD Drive: LG52X
• Keyboard: Logitech 110 Keys
• Mouse: Logitech Mouse

Software:
• Operating System: Windows 7
• Coding: Jupyter Notebook Code
• Browser: Microsoft Internet Explorer
• Environment: Node.Js

Programming Language used:
• Python
• Packages used: npm, pandas, Sequential, Dense, Lawet and LSTM

5 Results and Discussions

Even though there are many models already available in the market which serve the similar purpose of Predicting the Stock Price, ProfitPlus will be providing some more Efficiency on the top which have not been implemented yet or have been poorly implemented in the past. By combining experience from the past and knowledge from the present, an idea for this model has been processed, which has the potential of estimating the Google as well as Tesla stock price (Fig 10, Fig 11). This feature will not only provide investors to see the estimation for stock prices but also for beginners it will be a great model to see for estimating market share Price.

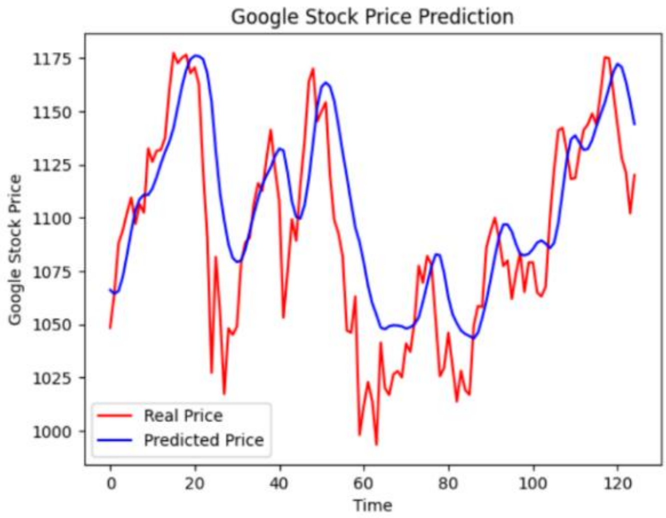

Fig 10 Estimated Stock Prices for Google stocks

Fig 11 Estimated Stock Prices for Tesla stocks

6 Conclusion

The creation of **ProfitPlus**, an RNN-based model designed to predict stock prices for major companies like Google and Tesla, represents a significant milestone in the application of machine learning to the financial sector. Stock price prediction is a highly complex and challenging task, given the multitude of variables that influence market movements. These include macroeconomic factors, investor sentiment, global events, and company-specific news, all of which contribute to the inherent volatility and unpredictability of stock prices. Despite these complexities, the development of a reliable model that can predict the stock prices of Google and Tesla with reasonable accuracy is a remarkable accomplishment. The model use of **Recurrent Neural Networks (RNNs)**, particularly Long Short-Term Memory (LSTM) networks, is an important factor in its success. LSTMs are capable of processing sequences of data and retaining long-term dependencies, making them well-suited for time-series forecasting tasks like stock price prediction. By learning from historical data, ProfitPlus can identify patterns and trends in the stock market, providing investors with valuable insights. While the model does not promise perfect accuracy due to the unpredictable nature of the market, its ability to deliver reliable predictions underlines the potential of machine learning to improve financial decision-making. However, stock markets are always evolving, and new challenges emerge as market dynamics change. Therefore, it is crucial for **ProfitPlus** to continually improve and adapt to these shifts. Regular updates and refinements to the model are necessary to maintain its effectiveness and ensure it remains relevant in a fast-changing financial landscape. This ongoing development will allow ProfitPlus to provide up-to-date, accurate predictions and remain a valuable tool for investors. In conclusion, the successful development of ProfitPlus highlights the transformative potential of technology in understanding complex systems like the stock market. By leveraging machine learning, platforms like ProfitPlus can offer investors data-driven insights that help them make informed decisions. As technology continues to advance, the role of AI and machine learning in financial markets will only grow, enabling more accurate predictions and better decision-making processes. The future of financial forecasting lies in the continued refinement and application of these powerful tools.

7 Declarations

7.1 Competing Interests

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

7.2 Publisher's Note

AIJR remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

How to Cite

Hemank Jain (2025). Smart Stock Price Prediction Algorithm using RNN variant Long Short-Term Memory. *AIJR Proceedings*, 1-11. https://doi.org/10.21467/proceedings.178.1

References

[1] Y. S. Kim, K. H. Lee, "Stock price movement estimation with lstm", neural network, Vol. 1 of the 2017 IEEE International Conference on Japanese Neural Networks (ICJNN 2017).

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," NeuralComput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available:http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[3] A. Graves, M. Liwicki, S. Fern´andez, R. Bertolami, H. Bunke, andJ. Schmidhuber, "A novel connectionist system for unconstrained hand-writing recognition," IEEE Transactions on Pattern Analysis and Ma-chine Intelligence, vol. 31, no. 5, pp. 855–868, May 2009.

[4] K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, andJ. Schmidhuber, "LSTM: A search space odyssey," arXiv preprintarXiv:1503.04069, 2015.

[5] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returnsprediction: A case study of china stock market," in Big Data (Big Data),2015 IEEE International Conference on, Oct 2015, pp. 2823–2824.

[6] O. H. Luca Di Persio, "Artificial neural networks approach to the forecastof stock market price movements," nternational Journal of Economicsand Management Systems, vol. 1, pp. 158–162, 2016.

[7] E. Hoseinzade and S. Haratizadeh, ''CNNpred: CNN-based stock market prediction using a diverse set of variables,'' Exp. Syst. Appl., vol. 129, pp. 273–285, Sep. 2019.

[8] S. Usmani and J. A. Shamsi, ''LSTM based stock prediction using weighted and categorized financial news,'' PLoS ONE, vol. 18, no. 3, Mar. 2023, Art. no. e0282234.

[9] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang, ''A stock closing price prediction model based on CNN-BiSLSTM,'' Complexity, vol. 2021, pp. 1–12, Sep. 2021.

[10] S. Siami-Namini, N. Tavakoli, and A. S. Namin, ''The performance of LSTM and BiLSTM in forecasting time series,'' in Proc. IEEE Int. Conf. Big Data, Dec. 2019, pp. 3285–3292.

[11] Y. Chen, R. Fang, T. Liang, Z. Sha, S. Li, Y. Yi, W. Zhou, and H. Song, ''Stock price forecast based on CNN-BiLSTM-ECA model,'' Sci. Program., vol. 2021, pp. 1–20, Jul. 2021.

[12] W. Lu, J. Li, J. Wang, and L. Qin, ''A CNN-BiLSTM-AM method for stock price prediction,'' Neural Comput. Appl., vol. 33, no. 10, pp. 4741–4753, May 2021.

[13] K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, andJ. Schmidhuber, "LSTM: A search space odyssey," S. Hochreiter and J. Schmidhuber, "Long short-term memory," NeuralComput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," NeuralComput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997.