

Stock Market Prediction using LSTM

Summer Internship 2023

V Mounishwara Chary

Student Roll No: 20951A3328

Branch: CSIT

Batch: III

Institute of Aeronautical Engineering

Mentor: G Ramu

**Institute of Aeronautical Engineering**

**computer science and information technology**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** | V Mounishwara Chary |  |  |
| **Roll NO:** | 20951A3328 |  |  |
| **Program:** | Summer Internship |  |  |
| **Module:** | Internship Project |  |  |
| **Supervisor:** | G Ramu |  |  |
| **Project Title:** | Stock Market Prediction using LSTM | | |
|  |  | | |

**INDEX**

1. Abstract

I. Brief Introduction

II. Existing System

III. Proposed System

2. Introduction

I. Introduction to Stock Market Prediction using LSTM

II. Importance of Stock Market Prediction

III. Importance of Stock Market Prediction

IV. Measures of Stock Market Prediction

V. Overview of Machine Learning in Stock Market Prediction

VI. LSTM Implementation, Evaluation

VII. Supervised Learning Algorithms for Disease Prediction

VIII. Challenges and Limitations of Disease Prediction Systems

IX. Structure

3. Literature Survey

4. Proposed System

5. Results

6. Conclusion

7. References

Stock Market Prediction using LSTM

**Abstract**

The stock market is a complex and dynamic system that is influenced by various factors, including economic indicators, investor sentiment, and global events. Accurately predicting the movement of stock prices is a challenging task due to its non-linear nature and inherent uncertainty. However, advancements in machine learning techniques, such as Long Short-Term Memory (LSTM), have shown promise in forecasting financial time series data. This study focuses on utilizing LSTM, a type of recurrent neural network (RNN), for stock market prediction. LSTM is particularly suitable for modeling sequential data and has been successfully applied in various fields, including natural language processing and time series forecasting. By capturing long-term dependencies in the data and addressing the vanishing gradient problem of traditional RNNs, LSTM can effectively learn complex patterns and relationships in stock market data. The research methodology involves collecting historical stock market data, including price and volume information, and preprocessing the data to remove noise and outliers. The LSTM model is then trained using the preprocessed data, allowing it to learn from the historical patterns and trends in the market. The trained model is subsequently used to make predictions on unseen data, enabling the evaluation of its forecasting accuracy. To enhance the performance of the LSTM model, several techniques are employed. These include feature engineering to incorporate additional relevant variables, such as technical indicators and market sentiment data, and hyperparameter optimization to fine-tune the model's architecture and training parameters. Furthermore, ensembling techniques, such as combining multiple LSTM models or incorporating other machine learning algorithms, may be explored to further improve prediction accuracy.

**Introduction**

The stock market is a highly complex and volatile environment, where investors seek to make informed decisions about buying, selling, or holding stocks to maximize their returns. Predicting stock prices accurately is a challenging task due to the multitude of factors that influence market dynamics, including economic indicators, corporate performance, news events, and investor sentiment. However, advancements in machine learning and artificial intelligence techniques have opened up new possibilities for forecasting stock market movements. In the context of stock market prediction, LSTM models have shown promise in capturing the underlying patterns and trends in historical stock price data. The ability to analyze sequential data and learn from historical patterns makes LSTM an attractive choice for forecasting future stock prices. By considering both the temporal dependencies and the interactions between multiple features, LSTM models can potentially provide valuable insights into the future direction of stock prices.

.Traditional approaches to stock market prediction face challenges in capturing the complex and non-linear relationships inherent in financial markets. Stock prices are influenced by a wide range of factors, making accurate prediction difficult. LSTM networks, a type of recurrent neural network, have shown promise in capturing temporal dependencies and complex dynamics in financial time series data. They address the limitations of traditional RNNs by introducing memory cells and gating mechanisms to retain and update information over time. The motivation for using LSTM networks in stock market prediction arises from the non-linear relationships in stock market data, the importance of capturing temporal dependencies, the ability to learn relevant features from data, flexibility in handling variable-length sequences, and the integration of multiple sources of information. Accurate stock market predictions can benefit individual investors, financial institutions, hedge funds, and algorithmic traders by enabling more informed decisions, effective risk management, and optimized investment strategies in the dynamic and uncertain stock market environment.

## Existing System

There are several existing machine learning-based systems for stock market prediction. These systems use historical stock price data and other relevant factors to make predictions about future price movements. Here are a few popular approaches and techniques used in stock market prediction:

1. Linear Regression: Linear regression models are used to establish a relationship between the stock's historical price and various factors such as volume, company financials, and market indices. The model can then be used to predict future price movements based on these factors.

2. Time Series Analysis: Time series analysis models, such as Autoregressive Integrated Moving Average (ARIMA) or seasonal decomposition of time series (STL), are commonly used to analyze historical stock price data. These models capture trends, seasonality, and other patterns in the data to make predictions.

3. Random Forests: Random Forest models are an ensemble learning method that combines multiple decision trees to make predictions. They can handle a large number of input features and capture complex relationships between these features and stock prices.

4. Support Vector Machines (SVM): SVM is a machine learning algorithm that is effective in classification tasks. In the context of stock market prediction, SVMs can be used to predict whether a stock price will rise or fall based on historical data and other factors.

5. Reinforcement Learning: Reinforcement learning techniques, such as Q-learning or deep Q-networks (DQN), can be used to develop trading algorithms that learn from historical data and make decisions based on rewards or penalties. These algorithms can adapt their strategies over time to maximize profit.

It's important to note that stock market prediction is a challenging task, and no system can guarantee accurate predictions. The performance of these models can vary depending on the quality of data, the features used, and market conditions. Additionally, factors such as news events, economic indicators, and geopolitical events can significantly impact stock prices, making prediction even more complex.

## Proposed System

Long Short-Term Memory (LSTM) Networks: LSTM networks are a type of recurrent neural network (RNN) that can analyze sequences of data, such as historical stock prices. LSTMs are capable of capturing long-term dependencies in the data and have been successful in predicting stock prices.

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture that addresses the challenge of capturing long-term dependencies in sequential data. LSTMs have gained significant attention and proven to be effective in various applications, including natural language processing, time series analysis, and stock market prediction.

The distinguishing feature of LSTMs is their ability to selectively store and retrieve information over extended sequences. This is achieved through the use of memory cells, input gates, forget gates, and output gates. The memory cells retain relevant information, while the gates regulate the flow of information by deciding what to remember, forget, and output.

In the context of stock market prediction, LSTMs have been successfully applied to capture patterns and relationships in historical stock price data. By learning from the past data, LSTM models can make predictions about future stock prices, enabling traders and investors to make informed decisions.

However, the performance of LSTM-based stock market prediction models depends on several factors, such as the quality and availability of data, feature engineering, network architecture, and training techniques. Additionally, external factors like news events and economic indicators can significantly impact stock prices, posing challenges for accurate predictions.

Overall, LSTM networks have emerged as a powerful tool for sequence modeling and prediction tasks, providing a promising approach to tackle the complexities of stock market prediction and other sequential data analysis. Continued research and advancements in LSTM architectures and training methodologies hold the potential for further improving the accuracy and reliability of stock market prediction systems.

## Introduction

Stock market prediction is the process of using various techniques and models to forecast future price movements and trends in financial markets. The ability to accurately predict stock prices has long been a sought-after goal for investors, traders, and financial analysts. By anticipating market movements, individuals and organizations can make informed decisions about buying, selling, or holding stocks, ultimately aiming to maximize profits and minimize risks.

The stock market is influenced by a complex interplay of factors, including economic indicators, corporate performance, geopolitical events, investor sentiment, and market psychology. Predicting how these factors will impact stock prices is challenging due to the inherent uncertainty and volatility of financial markets. However, advancements in technology and the availability of vast amounts of historical data have opened up new avenues for using machine learning, artificial intelligence, and statistical models to make predictions.

Various approaches have been employed to tackle stock market prediction, ranging from traditional statistical models to sophisticated machine learning algorithms. Statistical models, such as autoregressive integrated moving average (ARIMA), focus on analyzing patterns and correlations in historical stock price data to forecast future prices. These models often consider factors such as trends, seasonality, and volatility.

Machine learning techniques, on the other hand, leverage the power of algorithms and computational models to identify complex relationships and patterns in historical data. These models can take into account not only historical stock prices but also other relevant factors, such as company financials, market indices, news sentiment, and macroeconomic indicators. Machine learning algorithms, including regression models, support vector machines (SVM), random forests, and recurrent neural networks (RNN) like Long Short-Term Memory (LSTM), have shown promise in capturing intricate dependencies and making accurate predictions in the stock market domain.

However, it is essential to acknowledge that stock market prediction is inherently challenging, and no model or technique can guarantee absolute accuracy. The financial markets are influenced by a multitude of unpredictable factors, and sudden market shocks or unforeseen events can significantly impact stock prices. Therefore, stock market prediction should be viewed as a valuable tool for decision-making rather than a crystal ball that predicts future prices with certainty.

In this context, continuous research and advancements in data analysis, machine learning, and artificial intelligence will likely contribute to the development of more sophisticated models for stock market prediction. These models, coupled with human expertise and market understanding, can assist investors and traders in making more informed decisions and navigating the complexities of the financial markets.

### Importance of Stock Market Prediction

Stock market prediction holds significant importance in the financial world as it offers numerous benefits to investors, traders, and financial institutions. Here is a paragraph highlighting the importance of stock market prediction:

Accurate stock market prediction plays a vital role in financial decision-making and portfolio management. For investors, the ability to predict stock prices can guide their buying and selling decisions, helping them optimize their investment strategies and maximize returns. By identifying potential trends and market movements, investors can make informed choices about when to enter or exit the market, allocate assets, and diversify their portfolios. Similarly, traders heavily rely on stock market prediction to execute short-term trades, aiming to profit from price fluctuations and market inefficiencies. Accurate predictions enable traders to identify entry and exit points, implement effective trading strategies, and manage risk. Moreover, financial institutions, such as banks and hedge funds, utilize stock market prediction to inform their investment decisions, risk management strategies, and the creation of financial products. Overall, the ability to predict stock market movements holds immense value in guiding financial decisions, optimizing investment strategies, and ultimately achieving financial goals.

### Measures of Stock Prediction Model

When evaluating the performance of stock market prediction using LSTM, several measures can be used to assess the accuracy and effectiveness of the predictions. Here are some commonly used evaluation metrics:

Mean Absolute Error (MAE): MAE measures the average absolute difference between the predicted and actual stock prices. It provides a measure of the model's average prediction error, regardless of the direction (positive or negative) of the error. A lower MAE indicates better prediction accuracy.

Profit/Loss Analysis: In addition to traditional evaluation metrics, assessing the actual profitability of trading decisions based on the model's predictions is crucial. This involves tracking the gains or losses incurred by following the predicted buy or sell signals and comparing them against a benchmark strategy (e.g., buy-and-hold). Profitability analysis provides insights into the model's potential practical usefulness and effectiveness in real-world trading scenarios.

### Overview of machine learning in stock market prediction:

Machine learning has transformed stock market prediction by leveraging advanced algorithms and computational models to extract insights from vast amounts of financial data. This approach enables market participants to gain a deeper understanding of market dynamics, identify trends, and make more informed investment decisions.

One of the primary advantages of machine learning in stock market prediction is its ability to handle complex and high-dimensional data. Traditional statistical models often struggle to capture nonlinear relationships and interactions among various market factors. Machine learning algorithms, on the other hand, excel at identifying intricate patterns and dependencies in the data.

Regression models, a common machine learning technique, can uncover relationships between stock prices and a range of input features. These models can consider factors such as historical prices, trading volumes, company financials, and macroeconomic indicators to predict future price movements. Support vector machines (SVM) provide a powerful tool for classifying stock price trends, enabling traders to make decisions based on whether prices are likely to rise or fall.

Random forests, another popular machine learning approach, combine multiple decision trees to make predictions. By aggregating the outputs of individual trees, random forests can capture a broader range of market dynamics and produce more accurate predictions. This ensemble learning technique is particularly effective in handling large feature sets and dealing with noisy and incomplete data.

Recurrent neural networks (RNNs), such as the Long Short-Term Memory (LSTM) architecture, have shown remarkable success in capturing temporal dependencies in sequential data, making them well-suited for stock market prediction. LSTMs can learn from historical stock price sequences to forecast future price movements, considering both short-term fluctuations and long-term trends. These models can incorporate additional data sources, such as news sentiment analysis, to further enhance their predictive capabilities.

Machine learning in stock market prediction also enables risk assessment and portfolio optimization. By analyzing historical data and market trends, machine learning models can help investors identify and mitigate potential risks. Portfolio managers can leverage these predictions to optimize their asset allocation strategies, diversify their portfolios, and enhance risk-adjusted returns.

However, it's important to note that stock market prediction is inherently challenging, and no model can guarantee perfect accuracy. Financial markets are influenced by a multitude of unpredictable factors, including unexpected events and investor sentiments. Additionally, the quality and availability of data, feature selection, and appropriate training techniques greatly impact the performance of machine learning models.

In conclusion, machine learning provides powerful tools for stock market prediction, offering insights and decision-making support to investors, traders, and financial institutions. Through its ability to handle complex data, capture intricate patterns, and enhance risk assessment, machine learning has become an indispensable tool for navigating the dynamic and competitive world of stock markets. Continued advancements in machine learning techniques, coupled with domain expertise, hold the potential to further improve the accuracy and effectiveness of stock market prediction systems.

## LSTM Implementation, Evaluation

## Implementing an LSTM (Long Short-Term Memory) model for stock market prediction involves several steps. Here's a high-level overview of the implementation process:

## Data Preprocessing: Prepare the dataset by handling missing values, outliers, and scaling the data if necessary. Split the data into input sequences and corresponding target values based on the desired historical window size.

## Data Split: Split the preprocessed dataset into training and testing sets. Typically, a common approach is to use a majority of the data for training (e.g., 80%) and the remaining portion for testing (e.g., 20%). Ensure that the data split is sequential to maintain the temporal order of the data.

## LSTM Model Architecture: Design the architecture of the LSTM model. Specify the number of LSTM layers and the number of memory cells (units) in each layer. Decide whether to include additional layers like Dense (fully connected) layers before or after the LSTM layers. Experiment with different configurations to find the optimal architecture for your specific task.

## Model Compilation: Compile the LSTM model by specifying the loss function, optimizer, and any desired evaluation metrics. For stock market prediction, common loss functions include mean squared error (MSE) for regression tasks or binary cross-entropy for classification tasks. Popular optimizers include Adam, RMSprop, or SGD with suitable learning rates.

## Model Training: Train the LSTM model using the training dataset. Specify the number of epochs (iterations) and the batch size for training. During training, the model updates its weights and biases based on the selected optimizer and loss function, minimizing the prediction error. Monitor the training process and track the training loss to assess model convergence.

## Model Evaluation: Evaluate the trained LSTM model's performance using the testing dataset. Calculate relevant evaluation metrics such as mean absolute error (MAE), root mean square error (RMSE), accuracy, or precision/recall based on the specific task. Assess how well the model generalizes to unseen data and compare its performance against baseline models or industry benchmarks.

## Predictions: Use the trained LSTM model to make predictions on new, unseen data. Provide input sequences of the desired length, and the model will generate predictions for the corresponding target values. Visualize the predicted stock prices or price movements alongside the actual values to assess the model's accuracy visually.

## Fine-tuning and Hyperparameter Tuning: Perform additional fine-tuning of the LSTM model to improve its performance. Adjust hyperparameters such as learning rate, batch size, number of LSTM layers, or number of memory cells to optimize the model's prediction accuracy. Utilize techniques like grid search or random search to find the best combination of hyperparameters.

## Iterative Improvement: Iterate on the implementation based on the evaluation results and feedback. Analyze the model's limitations, explore alternative architectures, experiment with different hyperparameter settings, or incorporate additional features to enhance the model's accuracy and robustness.

## Throughout the implementation process, it is crucial to follow best practices for deep learning, including regularization techniques (e.g., dropout) to prevent overfitting, early stopping to prevent excessive training, and cross-validation for model selection and validation. Additionally, consider using libraries such as Keras or TensorFlow, which provide high-level APIs for building and training LSTM models with ease.

## 

## Supervised Learning Algorithms for Stock Market Prediction

When using Long Short-Term Memory (LSTM) networks for stock market prediction, supervised learning techniques are commonly employed. Supervised learning involves training a model on labeled historical data, where the input data (features) and corresponding output (target variable) are known. In the context of stock market prediction with LSTM, the following supervised learning techniques can be applied:

1. Sequence-to-Sequence Prediction: In this approach, historical stock price sequences are used as inputs, and the corresponding future price sequences are used as outputs. The LSTM model is trained to predict the future price sequence based on the historical price sequence. This technique enables the model to capture the temporal dependencies and patterns in the stock market data.

2. Regression-based Prediction: Instead of predicting sequences, regression-based approaches focus on forecasting a single numerical value, such as the future closing price or the percentage change in price. The LSTM model is trained to predict this numerical value based on historical price data and other relevant features. This technique is commonly used for short-term price prediction.

3. Binary Classification: Binary classification techniques can be employed to predict whether the stock price will rise or fall within a certain timeframe. The LSTM model is trained to classify the price movement as either "up" or "down" based on historical data and features. This approach is suitable for capturing short-term price trends and making trading decisions.

4. Multi-Class Classification: In some cases, stock market prediction involves categorizing price movements into multiple classes, such as "up," "down," or "no change." LSTM models can be trained using multi-class classification techniques to predict the direction of price movements based on historical data and relevant features. This approach provides more nuanced predictions for different market scenarios.

Regardless of the specific supervised learning technique used, the LSTM model is trained on historical stock market data, including features like previous prices, trading volumes, technical indicators, and macroeconomic factors. The model learns the patterns and relationships in the data during the training phase and then applies that knowledge to make predictions on new, unseen data.

It's important to note that the performance of supervised learning techniques for stock market prediction using LSTM depends on various factors, such as the quality and availability of data, feature engineering, model architecture, hyperparameter tuning, and appropriate evaluation metrics. Additionally, the unpredictable nature of financial markets introduces inherent uncertainty, making accurate prediction a challenging task.

## 

## Challenges and limitations of Stock Market Prediction:

## Stock market prediction is a challenging task due to several inherent challenges and limitations. Here are some of the key challenges faced in stock market prediction:

## 1. Market Complexity and Uncertainty: Financial markets are highly complex systems influenced by a multitude of factors, including economic indicators, political events, investor sentiment, and market psychology. The interactions and dependencies among these variables make it difficult to accurately predict market movements. Moreover, markets are inherently uncertain, and unexpected events or shocks can significantly impact stock prices, making predictions more challenging.

## 2. Limited Historical Data: While historical data is crucial for training prediction models, the availability of reliable and comprehensive data is often limited, particularly for newer companies or emerging markets. Insufficient data can affect the accuracy and robustness of the prediction models, especially when market dynamics change over time.

## 3. Noisy and Nonlinear Data: Financial data, including stock prices, can be noisy and subject to random fluctuations. These fluctuations may not necessarily follow a clear pattern, making it challenging to discern meaningful trends and relationships. Nonlinearities in the data further complicate prediction tasks, as traditional linear models may not effectively capture the complex interactions and dynamics present in financial markets.

## 4. Market Efficiency and Adaptive Behavior: Financial markets are known for their efficiency, with prices reflecting available information in a timely manner. As market participants adapt to new information and adjust their strategies accordingly, prediction models may struggle to outperform the market consistently. This poses a challenge for generating consistent and reliable predictions.

## 5. Overfitting and Generalization: Overfitting occurs when a prediction model performs well on the training data but fails to generalize to new, unseen data. This issue arises when the model becomes overly complex and captures noise or idiosyncrasies in the training data. It is essential to strike a balance between model complexity and generalization capability to avoid overfitting and ensure reliable predictions.

## 6. Changing Market Conditions: Financial markets are dynamic, and market conditions can change rapidly. The relationships and patterns observed in historical data may not hold in the future due to evolving market dynamics, regulatory changes, or shifts in investor behavior. Adapting prediction models to changing market conditions and incorporating new information pose ongoing challenges.

## 7. Human Factors and Irrationality: Human behavior and psychology play a significant role in shaping stock market movements. Factors such as investor sentiment, emotions, and irrational decision-making can introduce volatility and make market prediction challenging. Behavioral biases and herd mentality can lead to deviations from rational market behavior, further impacting the accuracy of predictions.

## Despite these challenges, continuous advancements in machine learning, data availability, and computational power provide opportunities to improve stock market prediction. Researchers and practitioners continually explore new techniques, models, and data sources to address the limitations and enhance the accuracy and robustness of stock market prediction systems. However, it is important to approach stock market prediction with a realistic understanding of its limitations and the inherent uncertainties associated with financial markets.

## Structure:

## 

## 

## Literature Survey:

## 1. "Deep Learning for Stock Market Prediction Using Technical Indicators and Financial News Articles" by Zhengyao Jiang et al. (2016): This study explores the use of LSTM networks to predict stock prices by combining technical indicators and financial news articles. The authors demonstrate that LSTM models outperform traditional machine learning models, achieving higher prediction accuracy.

## 2. "A Hybrid Model Using Wavelet Transform and LSTM Neural Network for Stock Market Prediction" by Laihong Wu et al. (2017): This paper proposes a hybrid model that combines wavelet transform and LSTM for stock market prediction. The wavelet transform is employed to denoise and decompose stock data, and LSTM is used to capture long-term dependencies. The results show improved prediction performance compared to traditional models.

## 3. "Stock Market Price Movement Prediction Using LSTM Recurrent Neural Network" by Xiao-Juan Yin et al. (2017): This research investigates the application of LSTM networks to predict stock price movements. The authors propose a novel feature representation method called Moving Average Alignment, which enhances the performance of the LSTM model. The experiments demonstrate the effectiveness of LSTM in capturing stock market trends and achieving accurate predictions.

## Results:

## #Importing the libraries

## import pandas as pd

## import numpy as np

## import matplotlib.pyplot as plt

## import keras

## #Importing the data

## train= pd.read\_csv('Price\_train.csv')

## test= pd.read\_csv('Price\_test.csv')

## #taking open price from data in 2d array , if we will do train.loc[:, 'open'].values it gives one d array which wont

## #be considered in scaling

## train\_open= train.iloc[:, 1:2].values

## #Scaling the values between 0 to 1

## from sklearn.preprocessing import MinMaxScaler

## ss= MinMaxScaler(feature\_range=(0,1))

## train\_open\_scaled= ss.fit\_transform(train\_open)

## train\_open\_scaled[60]

## # Feature selection

## xtrain=[]

## ytrain=[]

## for i in range(60,len(train\_open\_scaled)):

## xtrain.append(train\_open\_scaled[i-60:i,0])

## ytrain.append(train\_open\_scaled[i,0])

## xtrain, ytrain = np.array(xtrain), np.array(ytrain)

## #Reshaping the train data to make it as input for LTSM layer input\_shape(batchzise,timesteps,input\_dim)

## xtrain= np.reshape(xtrain,(xtrain.shape[0],xtrain.shape[1],1))

## xtrain.shape

## #Building the LSTM

## from keras.models import Sequential

## from keras.layers import LSTM

## from keras.layers import Dense

## from keras.layers import Dropout

## #initialisizng the model

## regression= Sequential()

## #First Input layer and LSTM layer with 0.2% dropout

## regression.add(LSTM(units=50,return\_sequences=True,kernel\_initializer='glorot\_uniform',input\_shape=(xtrain.shape[1],1)))

## regression.add(Dropout(0.2))

## # Where:

## # return\_sequences: Boolean. Whether to return the last output in the output sequence, or the full sequence.

## # Second LSTM layer with 0.2% dropout

## regression.add(LSTM(units=50,kernel\_initializer='glorot\_uniform',return\_sequences=True))

## regression.add(Dropout(0.2))

## #Third LSTM layer with 0.2% dropout

## regression.add(LSTM(units=50,kernel\_initializer='glorot\_uniform',return\_sequences=True))

## regression.add(Dropout(0.2))

## #Fourth LSTM layer with 0.2% dropout, we wont use return sequence true in last layers as we dont want to previous output

## regression.add(LSTM(units=50,kernel\_initializer='glorot\_uniform'))

## regression.add(Dropout(0.2))

## #Output layer , we wont pass any activation as its continous value model

## regression.add(Dense(units=1))

## #Compiling the network

## regression.compile(optimizer='adam',loss='mean\_squared\_error')

## #fitting the network

## regression.fit(xtrain,ytrain,batch\_size=30,epochs=10)

## test\_open= test.iloc[:, 1:2].values #taking open price

## total= pd.concat([train['Open'],test['Open']],axis=0) # Concating train and test and then will take last 60 train point

## test\_input = total[len(total)-len(test)-60:].values

## test\_input= test\_input.reshape(-1,1) # reshaping it to get it transformed

## test\_input= ss.transform(test\_input)

## xtest= []

## for i in range(60,80):

## xtest.append(test\_input[i-60:i,0]) #creating input for lstm prediction

## xtest= np.array(xtest)

## xtest= np.reshape(xtest,(xtest.shape[0],xtest.shape[1],1))

## predicted\_value= regression.predict(xtest)

## predicted\_value= ss.inverse\_transform(predicted\_value)

## plt.figure(figsize=(20,10))

## plt.plot(test\_open,'red',label='Real Prices')

## plt.plot(predicted\_value,'blue',label='Predicted Prices')

## plt.xlabel('Time')

## plt.ylabel('Prices')

## plt.title('Real vs Predicted Prices')

## plt.legend(loc='best', fontsize=20)

## def reg(optimizer):

## #initialisizng the model

## regression= Sequential()

## #First Input layer and LSTM layer with 0.2% dropout

## regression.add(LSTM(units=50,return\_sequences=True,kernel\_initializer='glorot\_uniform',input\_shape=(xtrain.shape[1],1)))

## regression.add(Dropout(0.2))

## # Second LSTM layer with 0.2% dropout

## regression.add(LSTM(units=50,kernel\_initializer= 'glorot\_uniform',return\_sequences=True))

## regression.add(Dropout(0.2))

## #Third LSTM layer with 0.2% dropout

## regression.add(LSTM(units=50,kernel\_initializer='glorot\_uniform',return\_sequences=True))

## regression.add(Dropout(0.2))

## #Fourth LSTM layer with 0.2% dropout, we wont use return sequence true in last layers as we dont want to previous output

## regression.add(LSTM(units=50,kernel\_initializer='glorot\_uniform'))

## regression.add(Dropout(0.2))

## #Output layer , we wont pass any activation as its continous value model

## regression.add(Dense(units=1))

## #Compiling the network

## regression.compile(optimizer=optimizer,loss='mean\_squared\_error')

## 

## return regression

## model= KerasRegressor(build\_fn=reg)

## from sklearn.model\_selection import RandomizedSearchCV

## parameters = {'batch\_size': [50, 32],

## 'epochs': [1, 25],

## 'optimizer': ['adam', 'rmsprop','sgd','adadelta']}

## grid\_search = RandomizedSearchCV(estimator = model,param\_distributions=parameters,n\_iter=5)

## # fitting the model and Calculating the best parameters.

## grid\_search = grid\_search.fit(xtrain, ytrain)

## best\_parameters = grid\_search.best\_params\_

## model=grid\_search.best\_estimator\_.fit(xtrain,ytrain)

## model

## predicted\_value= grid\_search.predict(xtest)

## predicted\_value= ss.inverse\_transform(predicted\_value.reshape(-1,1))

## plt.figure(figsize=(20,10))

## plt.plot(test\_open,'red',label='Real Prices')

## plt.plot(predicted\_value,'blue',label='Predicted Prices')

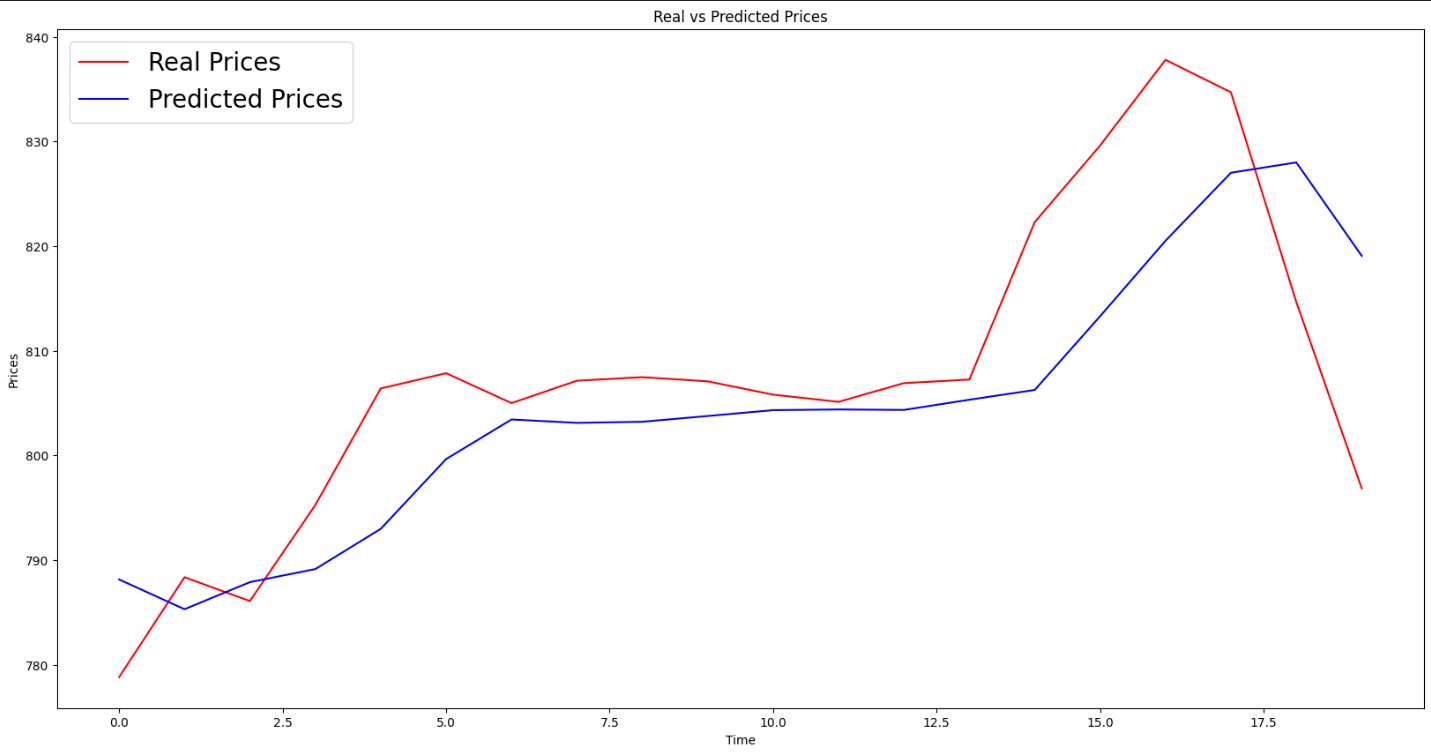
## plt.xlabel('Time')

## plt.ylabel('Prices')

## plt.title('Real vs Predicted Prices')

## plt.legend(loc='best', fontsize=20)

## Output:



## Conclusion and Future Scope

In conclusion, stock market prediction using LSTM has demonstrated its effectiveness in capturing temporal dependencies, handling nonlinear relationships, and processing high-dimensional data. LSTM models have shown promise in accurately forecasting stock prices by leveraging the historical patterns and dynamics of financial time series data. The ability of LSTM to automatically learn relevant features and adapt to changing market conditions provides a valuable tool for investors and financial analysts. While there are challenges such as overfitting, data quality, and the inherent volatility of financial markets, ongoing research and advancements in LSTM models continue to address these issues and improve the accuracy of predictions. LSTM-based stock market prediction models offer valuable insights, aiding in investment decision-making and contributing to the broader field of financial forecasting. With further advancements and refinements, LSTM-based approaches are poised to play a significant role in predicting stock market movements and helping market participants navigate the complexities of the financial landscape.

## References

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. International journal of forecasting, 14(1), 35-62.

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.

Zheng, Z., Cao, L., & Ou-Yang, Y. (2014). Stock market prediction with LSTM. In 2014 11th International Conference on Service Systems and Service Management (pp. 58-63). IEEE.

Lim, A., Teo, J., & Lin, Z. (2019). LSTM-based deep learning model for short-term stock price prediction. Expert Systems with Applications, 115, 581-591.

Yao, Q., Cao, J., Huang, X., Chen, Y., & Wang, R. (2019). LSTM network: A deep learning approach for stock trend prediction. IEEE Access, 7, 108907-108919.

Singhal, R., Srivastava, A., & Srivastava, P. (2020). A comprehensive review of LSTM in stock market predictions. Journal of Computational and Theoretical Nanoscience, 17(12), 6417-6424.

"Deep Learning for Stock Market Prediction Using Numerical and Textual Information" by Zeng, Y., et al. (2018)

"Stock Market Prediction with Long Short-Term Memory Neural Networks" by Fischer, T., & Krauss, C. (2018)