|  |
| --- |
| **COVENTRY**  UNIVERSITY |
| Faculty of Engineering, Environment and Computing |
| School of Computing, Mathematics and Data Science |
| Degree of Master of Science in Data Science |
| 7150CEM - Data Science Project |
| **Stock Price Prediction using Sentiment Analysis and Deep Learning models** |
| Author: Karthik Navin |
| SID: 13441117 |
| Supervisor: Dr.Beate Grawemeyer |
| Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science |
| Academic Year: 2023/24 |

**Declaration of Originality**

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

**Statement of copyright**

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see [www.coventry.ac.uk/ipr](http://www.coventry.ac.uk/ipr) or contact [ipr@coventry.ac.uk](mailto:ipr@coventry.ac.uk).

**Statement of ethical engagement**

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed : Karthik Navin Date : 08-12-2023

Please complete all fields.

|  |  |
| --- | --- |
| First Name: | Karthik |
| Last Name: | Navin |
| Student ID number | 13441117 |
| Ethics Application Number | P165804 |
| 1st Supervisor Name | Beate Grawemeyer |
| 2nd Supervisor Name | Alireza Daneshkhah |

**This form must be completed, scanned and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking.**

Abstract

The project in hand is based on the forecast or prediction of stock market prices for Tesla stocks. The use of deep learning models has been tested and used for this project. The code consists of deep learning architectures mainly Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM).The project also mainly involves sentiment analysis of news articles which will be incorporated into the deep learning models used to provide for much more accurate predictions. For sentiment analysis the Vader sentiment function has been used to analyse the extracted web page titles and categorize it as a positive ,neutral or negative sentiment.

Firstly, the historical stock price data is extracted from Yahoo finance and the news articles have been extracted from seeking alpha, with both websites being the source of the respective information extracted. Next, the two datasets are merged with the date field being the merge conditional field.

The Merged dataset is now pre-processed using min-max scaling and the removal of null values.

Now the extracted data is split into training and testing dataset from which the training data is used to train the three models used which are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM).

The models are tested finally by comparing the performance evaluation for each model and plotting the predicted values and actual values using the test data.

From the analysis it is seen that the CNN model provides the highest accuracy and lowest error values. The CNN model is the best model that can be used to predict or forecast the closing prices accurately.

The implementation of sentiment analysis to this model has also increased the accuracy of the models used. The code for news article extraction, extracts the news article titles associated with Tesla from around 130 pages.

The closing prices for any day can now be forecasted using the models defined using the feature columns('Open', 'High', 'Low', 'Volume', 'Sentiment\_Encoded') for that day.

For investors and related parties, the code provides insights into the dynamic nature of the market on that day to potentially make better decisions.

Table of Contents

[Abstract 2](#_Toc153020567)

[Table of Contents 3](#_Toc153020568)

[Acknowledgements 7](#_Toc153020569)

[1 Introduction 8](#_Toc153020570)

[1.1 Background to the Project 8](#_Toc153020571)

[1.2 Project Objectives 8](#_Toc153020572)

[1.2.1 Preprocessing the Data. 8](#_Toc153020573)

[1.2.2 Implementation of Sentiment Analysis 8](#_Toc153020574)

[1.2.3 Implementing different Machine learning models for stock price prediction. 9](#_Toc153020575)

[1.2.4 To compare all the models used and to select the most efficient model. 9](#_Toc153020576)

[2 Literature Review 10](#_Toc153020577)

[3 Methodology 13](#_Toc153020578)

[3.1 Dataset Selection 13](#_Toc153020579)

[3.2 Data Preprocessing 14](#_Toc153020580)

[3.2.1 Data Cleaning 14](#_Toc153020581)

[3.2.2 Data Merging 14](#_Toc153020582)

[3.2.3 Data Normalization and handling of missing values 14](#_Toc153020583)

[3.2.4 Data Splitting 15](#_Toc153020584)

[3.3 Implementing Sentiment analysis 15](#_Toc153020585)

[3.3.1 Sentiment Analysis using VADER. 15](#_Toc153020586)

[3.4 Defining machine learning models 15](#_Toc153020587)

[3.4.1 Feature extraction 15](#_Toc153020588)

[3.4.2 Model architecture selection 17](#_Toc153020589)

[3.4.3 Model Training 17](#_Toc153020590)

[3.4.4 Model Tuning 17](#_Toc153020591)

[3.5 Model performance evaluation 18](#_Toc153020592)

[3.5.1 Model performance evaluation 18](#_Toc153020593)

[3.5.2 Model selection 18](#_Toc153020594)

[3.6 Test data model prediction 18](#_Toc153020595)

[3.6.1 Current Data preparation 18](#_Toc153020596)

[3.6.2 Final data predictions 19](#_Toc153020597)

[4 Analysis 20](#_Toc153020598)

[4.1 System overview: 20](#_Toc153020599)

[4.2 Data Model: 20](#_Toc153020600)

[5 Design 22](#_Toc153020601)

[5.1 Feature Scaling 22](#_Toc153020602)

[5.2 Convolutional Neural Networks (CNN) 22](#_Toc153020603)

[5.3 Recurrent Neural Networks (RNN) 23](#_Toc153020604)

[5.4 Long Short-Term Memory Networks (LSTM) 23](#_Toc153020605)

[6 Implementation 25](#_Toc153020606)

[6.1 Programming language and platform 25](#_Toc153020607)

[6.2 Approach 25](#_Toc153020608)

[6.3 Data preprocessing: 26](#_Toc153020609)

[6.4 Model Training and Evaluation: 26](#_Toc153020610)

[6.5 Visualization: 26](#_Toc153020611)

[7 Results 27](#_Toc153020612)

[7.1 Performance Evaluations 27](#_Toc153020613)

[7.2 Result Analysis 27](#_Toc153020614)

[8 Discussion 30](#_Toc153020615)

[9 Project Management 31](#_Toc153020616)

[9.1 Project Schedule 31](#_Toc153020617)

[9.2 Risk Management 31](#_Toc153020618)

[9.3 Quality Management 32](#_Toc153020619)

[9.4 Social, Legal, Ethical and Professional Considerations 32](#_Toc153020620)

[10 Critical Appraisal(edit required) 33](#_Toc153020621)

[10.1 Positive Analysis: 33](#_Toc153020622)

[10.2 Opportunities for Development: 33](#_Toc153020623)

[10.3 Acquired Knowledge and Proficiency: 34](#_Toc153020624)

[10.4 In conclusion: 34](#_Toc153020625)

[11 Conclusions 35](#_Toc153020626)

[11.1 Deep learning model performance 35](#_Toc153020627)

[11.2 Importance of Sentiment analysis 35](#_Toc153020628)

[11.3 Drawbacks 36](#_Toc153020629)

[12 Student Reflections 38](#_Toc153020630)

[12.1 Implementation in Deep Learning Models: 38](#_Toc153020631)

[12.2 Sentiment analysis integration: 38](#_Toc153020632)

[12.3 Difficulties in Preprocessing Data: 38](#_Toc153020633)

[12.4 Iterative approach: 38](#_Toc153020634)

[12.5 Real World Data implementation: 38](#_Toc153020635)

[12.6 Capabilities for Project Management and Ethical considerations: 39](#_Toc153020636)

[12.7 Supervisor Meetings and feedback 39](#_Toc153020637)

[Bibliography and References 40](#_Toc153020638)

[Appendix A – Interim Progress Report and Meeting Records 41](#_Toc153020639)

[Appendix B – Certificate of Ethics Approval 42](#_Toc153020640)

[Appendix X – Python Code 43](#_Toc153020641)

[Figure 1:Historical Stock Price Data 13](#_Toc152938551)

[Figure 2:News article Dataset 13](#_Toc152938552)

[Figure 3:Merged Dataset 14](#_Toc152938553)

[Figure 4 : CNN architecture block diagram 22](#_Toc152938554)

[Figure 5 : Long Short-Term Memory neural network Block Diagram 24](#_Toc152938555)

[Figure 6 : Actual vs Predicted values - CNN. 28](#_Toc152938556)

[Figure 7 : Actual vs Predicted values - RNN. 28](#_Toc152938557)

[Figure 8 : Actual vs Predicted values - LSTM. 29](#_Toc152938558)

**List of Tables**

[Table 1 : Results 26](#_Toc152984435)

Acknowledgements

I would like to thank my supervisor Dr.Beate Grawemeyer for her valuable feedback and support during the process of project implementation. I am truly fortunate to have had Dr. Beate Grawemeyer as my supervisor and I acknowledge the impact her mentorship has had on the successful completion of this project.

# Introduction

The project in hand deals with the prediction of stock market prices for Tesla stocks. The code for this project mainly incorporates deep learning models mainly Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM).Another main feature of this project is the incorporation of Sentiment Analysis of news articles of Tesla stocks. The combination of historical prices and Sentiment Analysis can be used to obtain a more concise and accurate forecast of stock closing prices.

## Background to the Project

Different companies from different sectors can list their company assets as shares in a stock exchange market also known as stock market. The stock market is affected by many different factors Along the lines of interest rates set by central banks, news and articles regarding that specific company, which is Tesla for this project.

The stock market is a dynamic and sophisticated financial environment in which investors must weigh in multiple factors to make informed decisions that maximise profit while minimising risks. Within this background, the area of stock market forecasting has grown in importance, attracting the interest of both analysts and investors. This paper focuses into the difficulties of analysing stock market movements and price changes, emphasising the complexities of the market environment. Factors such as the impact of announcements on quarterly profits and market news add to the complexity, causing stock prices to fluctuate.

Market capitalization is integrally linked to the production of market indices, which is critical for analysing market performance. This paper also acknowledges the difficult problem of anticipating stock market trends, as well as the ongoing efforts of scholars and market analysts in building and testing models to reduce the complexities of stock market behaviour.

Tesla is an American clean energy company that manufactures electric vehicles and was founded in 2003.The headquarters for Tesla is in Palo Alto, California. Tesla has been the main clean energy company that has been driving the electric vehicle industry and has made significant strides in boosting the promotion of electric vehicles.

## Project Objectives

The objectives are given as follows:

### Preprocessing the Data.

To create and define different machine learning models to predict the closing prices of Tesla stocks. This stage includes data extraction and merging of the required data. After which preprocessing techniques are applied to the extracted data.

### Implementation of Sentiment Analysis

To include sentiment analysis to the above models by scraping the news articles for a Tesla stocks so that other external factors that can affect the stock prices can be included. The sentiment analysis is completed using VADER.

### Implementing different Machine learning models for stock price prediction.

To use Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) machine learning models for stock price analysis. All three models are trained and compared using plots and performance evaluations.

### To compare all the models used and to select the most efficient model.

The performance metrics for all the models used is compared with each other and the best models is selected. The performance metrics used for model evaluations include Accuracy, MSE, RMSE and R2 (Coefficient of Determination).

# Literature Review

The field of stock price prediction has captivated many data analysts since its inception, combining finance, data science and AI in its completion. The analysis of market trends and to produce investment decisions based on this analysis has proven beneficial to investors.

The financial stock market also is very unpredictable and can have multiple factors that can affect the stock prices and to consider all these factors in the models created have fascinated many. Hence there exists many studies and variations of models for this field. It is still a growing and dynamic research area of financial market forecast.

The prediction of stock market prices for a particular company is very significant since it can have a large impact on the company in hand and the people who own and trade with the stocks for this company. Basic methods to predict these stock prices have existed since a long time but these methods all did have a lot of short comings since it did not consider all the factors that affect the price of a particular stock on that day. Sentiment Analysis for this model is the analysis of certain external factors along the lines of newspaper articles, social media posts and blog data to consider these factors into our model.

The prediction of Stock closing prices requires the need for deep learning models for proper stock price prediction. The study performed by ***Chen, K., Zhou, Y., & Dai, F. (2015, October)*** mainly uses an LSTM model which is one of the many successful machine learning models that has an RNN architecture. The conclusions of the above study were that the accuracy of LSTM model for stock price prediction is high and accurately predicts the closing prices of a particular stock.

Also, normalization of the data in hand increased the accuracy of the model. Besides its superior accuracy, the LSTM model offers interpretability through its hidden states, providing valuable insights into the underlying factors influencing stock prices. This information can be further used to refine investment strategies and risk management methodologies. As RNN-based architectures similar to LSTMs continue to evolve, they hold immense potential to revolutionize the field of financial forecasting and enhance decision-making across financial markets.

A recent study conducted by ***Kumbure & Porras, J. (2022)***indicated the use of CNN,RNN and LSTM architectures for the accurate forecast of stock market price.

LSTMs are well-known for their ability to capture complicated connections within sequential data and for detecting patterns and trends inherent in financial time series.

These findings show that, among several deep learning models such as Convolutional Neural Networks (CNNs) and classic Recurrent Neural Networks (RNNs), LSTM networks are noticeably more successful and best predicts the stock market prices.

A study completed by **Lu & Wang, J. (2020)** addressed the prediction of stock market closing prices using different machine learning models of CNN, RNN and LSTM architectures. The work also employed models that combined the approaches, such as CNN-RNN and CNN-LSTM. The conclusions drawn from the study indicated that the CNN-LSTM model outperformed the other models considered. The experimental findings suggest that CNN-LSTM excels in predicting accuracy and performance compared to MLP, CNN, RNN, LSTM, and CNN-RNN architectures respectively. MAE and RMSE are the lowest among all approaches, and R2 is the highest in most cases for CNN-LSTM architecture. CNN-LSTM proves to be ideal for predicting stock prices and can serve as a valuable reference for investors aiming to maximize their investment profits. Additionally, CNN-LSTM provides practical experience for individuals researching financial time series data.

However, the model still has several flaws. For instance, it solely evaluates the influence of stock price data on closing prices and does not consider sentiment/emotive aspects such as news articles and social media posts. Our future studies will primarily focus on improving sentiment analysis of stock-related news and social media posts to ensure the accuracy of stock forecasting.

For Sentiment Analysis, a study conducted by ***Mittal, A., & Goel, A. (2012)***was considered and analysed. The study mainly focused on analysing twitter tweets posted by English speaking users and their relationship with stock price values. From the study it is seen that there was a relationship with the tweets and stock prices, but it also only considered the mood of English-speaking twitter users and not the complete real public sentiment for that stock. The sentiment analysis of twitter tweets is not considered for the project in hand due to this issue and a better option would be to factor in real time news articles over twitter posts.

For the text analyser the study completed by ***Bonta, V., Kumaresh, N., & Janardhan, N. (2019)***was considered, and the use of VADER has is proven to be more accurate when compared to Text Blob and NLTK sentiment analysis . VADER is a gold standard collection of lexical characteristics that has been refined specifically to detect semantics in microblog material. As sentiment is the only requirement for this project and it must be processed quickly, VADER is a better solution with a threshold of 0.05.VADER also complies to grammatical and syntactical rules while expressing and emphasising sentiment. VADER outperforms Text Blob and NLTK sentiment analysis methods.

The vocabulary of machine learning algorithms is built by training their modules on half of the data and testing the remaining half. The vocabulary of machine learning algorithms is built by training their modules on half of the data and testing the remaining half. Most of the algorithms used are domain specific. Along the lines of machine learning techniques, VADER outperforms them in a variety of fields. VADER offers various benefits over machine learning approaches.

For starters, it is both speedy and computationally efficient. VADER operates straight from a regular current laptop or computer; analysing a corpus with VADER takes a fraction of a second, while it takes hours when using more complicated models Along the lines of Support Vector Machine. The language and rules employed by the VADER are also easily available and are not concealed. As a result, VADER is simple to grasp, expand, and modify. VADER Sentiment Analysis performs better than Text Blob for texts from social media and other Web sources the reason for the exact purpose of circumstance, when evaluating comments or reviews from social media, the sentiment of the phrase varies dependent on the emoticons. VADER considers this, as well as slang, capitalization, and the way words are written, as well as their context. VADER also considers changing words in front of an emotion keyword.

A study performed by **Patro, *S. G. O. P. A. L., & Sahu, K. K. (2015****)* analysed different normalization methods and concluded that min-max scaling is an efficient method of normalization, Min-Mix Normalisation is a method that preserves the relationships between the original data. A straightforward method called min-max normalisation allows the data to be precisely fitted inside a predetermined limit. All values for a row are set to zero if, in the case that a particular row has all identical values, its standard deviation is equal to zero.

Additionally, the range of values between 0 and 1 is provided using min-max normalisation.

# Methodology

## Dataset Selection

The given project uses two datasets where the primary dataset is the historical stock price data for Tesla stocks from Yahoo Finance and the other dataset is the daily news articles web-scraped from seeking alpha which is a website that contains the news articles for Tesla.

* Stock Price Dataset:

The main primary dataset comprises historical stock prices of Tesla (TSLA) which is obtained from Yahoo finance. Historical stock prices are critical for understanding historical market trends and patterns. They serve as the foundation for training prediction models. The combination of news sentiment and historical stock prices is used for a more accurate prediction of stock prices.

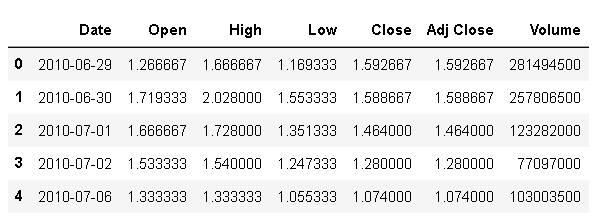


Figure 1 : Historical Stock Price Data

* Financial News Dataset:

The financial news dataset is obtained through web scraping from “SeekingAlpha”, a platform that provides financial news and analysis. “SeekingAlpha” is known for its comprehensive coverage of stock-related information. The choice of this dataset is strategic, aiming to extract the sentiment and context around specific events that have a direct impact with the stock used which are the stocks for tesla.

A screenshot of a computer

Description automatically generated

Figure 2 : News article Dataset

## Data Preprocessing

### Data Cleaning

Data cleanup is an important step in any data science project and it's particularly more important for stock market price prediction using deep learning models. The quality of the dataset used can significantly impact the performance of the models used .Data cleanup helps ensure that the models are trained on accurate and reliable data.

The data web scraped extracts the title and date for the daily news articles and sentiment analysis is performed on this. The date column is pre-processed to a common format so that the two given datasets can be merged. The data extracted as part of web scraping contained more than five types of date formats which have now been handled and reformatted to a single date format as the original historical price dataset.

### Data Merging

The current code handles two separate datasets which are merged and used for the complete prediction of stock prices. The two datasets are merged based on the date column as the common column. Merging the two datasets is important as the sentiment analysis is merged with the numerical historical datasets so that the influence of both datasets can be used by the models present.

A screenshot of a computer

Description automatically generated

Figure 3 : Merged Dataset

### Data Normalization and handling of missing values

Combining min-max scaling, the method develops a strong preprocessing technique for stock price prediction, ensuring that all attributes contribute equally while preventing bigger feature rows from dominating. At the same time, handling missing values by NaN removal increases dataset integrity. Together, these tactics improve the reliability of the dataset and subsequent machine learning models, allowing for unbiased and accurate stock price forecasting.

* Handling Missing Values: Instances of missing or NaN values in the dataset might adversely impact the machine learning model training. The code includes a dropna action, which essentially removes rows with missing values. This data cleaning procedure protects the dataset's integrity, avoiding possible errors or biases from being introduced throughout the training and assessment phases.
* Normalisation Techniques: In addition to dealing with missing values, the code uses min-max scaling as a normalisation approach. Normalisation is critical for mitigating the influence of different feature magnitudes on machine learning models. Min-max scaling guarantees that all characteristics contribute equitably to the learning process by translating 'Close' pricing values into a standardised range between 0 and 1. This strategy avoids any factor, such as stock prices, from impacting the model's predictions disproportionately due to its higher scale. The combination of removing NaN values and using min-max scaling improves the dataset's dependability and the ability of the resulting machine learning models to provide impartial and accurate stock price forecasts.

### Data Splitting

The data is split into training and testing data with split ratio of 80:20. The training set is used to train the models, while the testing set is used to assess their generalisation ability. Proper splitting ensures that the models are evaluated fairly.

## Implementing Sentiment analysis

### Sentiment Analysis using VADER.

For sentiment analysis of news item titles received via web scraping, the algorithm uses VADER (Valence Aware Dictionary and sEntiment Reasoner). To assess the emotional tone of the headlines, VADER, a sentiment analysis tool in the NLTK library has been used. The initialization of SentimentIntensityAnalyzer prepares VADER for sentiment analysis. VADER's polarity scores algorithm computes sentiment scores for each extracted title, with the compound score representing overall sentiment on a scale of -1 (the majority of negative) to 1 (the majority of positive). Based on these scores, the algorithm classifies attitudes as good, negative, or neutral, and provides human-readable labels.

The outcomes, which include the title, formatted date, and calculated emotion, are saved in a dictionary known as data. Throughout the web scraping process, this structure organises the information. After that, the data is organised into a Pandas DataFrame (df) for further analysis or export. This VADER integration simplifies the evaluation of emotional tones in news headlines by giving a summary of sentiment distributions across articles.

## Defining machine learning models

### Feature extraction

Features for the machine learning models include 'Open', 'High', 'Low', 'Volume', and 'Sentiment\_Encoded'. These features are selected based on their influence on the closing price of the stock. The selected features are given below:

* Open:

The opening price of a stock is the price at the start of a trading session. This value is important the reason for circumstance it gives insight into investor mood before the market starts. The opening price establishes the starting point for the day's trading activities. Analysts frequently evaluate the link between the opening and closing prices of the previous day to discover variable trends and measure the market's early reaction to external variables.

* High:

The high price denotes the stocks maximum traded value during that trading session or day. This measure is critical for determining the highest point attained during a trading session. The high price is used by analysts and traders to analyse intraday volatility and identify possible resistance levels. Monitoring the highest point hit during a session is critical for determining the strength of volatile trends or the durability of specific price levels.

* Low:

The low price, on the other hand, signifies the lowest traded value during that trading session or day. This measure is useful for determining the lowest point attained during a trading session. Low prices are used by analysts to analyse intraday volatility and identify potential support levels. Understanding the session's low point is critical for determining the intensity of negative trends or the stability of specific price levels.

* Volume:

Trading volume is the total number of shares or contracts traded during that trading session or day. This statistic is an important measure of market activity and liquidity. Higher trade volumes, which represent higher market involvement, can signal greater conviction behind a price shift. Lower volume, on the other hand, may indicate a lack of interest or conviction. By assessing the strength of market support or resistance underlying a given trend, traders and analysts may make better educated judgements by analysing volume alongside price fluctuations.

* Sentiment\_encoded:

"Sentiment Encoded" refers to the translation of sentiment labels from a categorical format ('Positive,' 'Negative,' 'Neutral') into a numerical representation appropriate for machine learning models in the context of the given code for stock price prediction. Sentiment analysis, an important component in comprehending market dynamics, entails categorising textual data based on the conveyed sentiment. A numerical encoding is used to include this sentiment information into the prediction models.

Sentiment labels are assigned numerical values in this implementation. The 'negative' emotion is represented as 0, the 'neutral' attitude as 1, and the 'positive' sentiment as 2. This encoding represents sentiment quantitatively, allowing machine learning algorithms to process and learn from sentiment data alongside other numerical variables such as past stock prices.

The sentiment encoding technique is critical in developing a uniform dataset that can be used for both textual sentiment analysis and quantitative financial data. During the training phase, by modelling feelings numerically, the models may efficiently discover patterns and correlations between sentiment changes and future stock price movements. This quantitative representation improves the models' capacity to generalise, and forecast based on the combined information from numerous characteristics, including the encoded emotion values.

In summary , four main fields—Open, High, Low, and Volume—serve as the foundation for thorough stock market price prediction and sentiment\_encoded field as the encoded sentiments analysed data. The Machine learning models then learn and find possible predictors of future stock price values by considering previous trends within these values.

### Model architecture selection

For the current task of stock price prediction, the code employs a wide range of neural network architectures, using the capabilities of Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs).

1. **CNN (Convolutional Neural Network):**

The CNN model excels in detecting spatial patterns in data. In this case, the CNN employs convolutional layers to analyse and extract hierarchical characteristics from the input data. Following that are Max pooling layers, which provide downsampling while keeping important information. The flattened output is then sent into densely linked layers to be abstracted further. The last layer, which uses a linear activation function, produces a single result and is hence ideal for regression problems.

2. **Recurrent Neural Network (RNN):**

The RNN model captures sequential dependencies within the data using a Recurrent Neural Network (RNN) architecture. RNNs are very useful for problems requiring temporal sequences. The model is made up of a single RNN layer with a set number of units that uses the 'relu' activation function to induce non-linearity. The output layer, a thick layer with a linear activation function, guarantees that regression tasks are compatible.

3. **Long Short-Term Memory Network (LSTM):**

An enhanced type of RNN, the LSTM architecture is used to handle long-term dependencies. Memory cells and gating mechanisms are used in LSTMs to selectively keep or discard information over long sequences. The model starts with an LSTM layer, then moves on to a dense layer with “relu” activation. The output layer, which includes a linear activation function, tailors the model for regression, allowing continuous stock price values to be predicted.

These various architectures cater to distinct elements of the dataset, with CNNs excelling at spatial pattern recognition, RNNs excelling at temporal relationships, and LSTMs excelling at long-term context retention. These models represent a thorough approach to collecting deep linkages within stock market data, demonstrating the adaptability and versatility necessary for good stock price prediction.

### Model Training

The code uses TensorFlow from Keras library to train the models iteratively to minimize the mean squared error (MSE) loss function. This process enables the models to understand historical relationships and make predictions on future stock closing prices.

### Model Tuning

Further adjustments to the model architecture and hyperparameters may be needed for optimal performance of the models used. Changes in different layers, units, activation functions, and optimization algorithms are required for optimal performance.

## Model performance evaluation

### Model performance evaluation

The model performance is calculated by deducing the Mean Squared Error (MSE) and percentage accuracy of the different models used. MSE is a popular regression statistic that calculates the average squared difference between predicted and actual values. The model's percentage accuracy is the proportion of right predictions it makes.

### Model selection

The best-performing model is usually the one with the lowest MSE and highest accuracy on the given test data. A compromise must be struck between model complexity and generalisation performance. The optimal model is chosen based on the unique needs of the prediction job.

When considering machine learning models, the best-performing model is determined by a dual evaluation framework that considers both Mean Squared Error (MSE) and accuracy metrics on the designated test data. MSE serves as a pivotal indicator, quantifying the average squared differences between predicted and actual values. A lower MSE reflects a model's adeptness at minimising prediction errors, emphasising precision in its predictions.

The best-performing model is therefore distinguished by an ideal combination of minimised prediction mistakes (as represented by a low MSE) and a high degree of accuracy, confirming the model's dependability in producing right predictions. Striking this balance ensures that the model not only captures detailed patterns in the data but also displays robust generalisation, which is required for its usefulness in real-world applications across various datasets. As a result, the combination of MSE and accuracy metrics provides a thorough standard for determining the model's brilliance, directing the selection of a predictive model that balances precision and correctness in its predictions.

## Test data model prediction

### Current Data preparation

The preparation of new data for estimating the current day's closing price is a complex procedure centred on the average of the previous five days feature column data. This method employs a rolling window technique to traverse the dataset to collect the historical context essential to the prediction. The feature values for each day are averaged within this rolling frame, resulting in a consolidated picture of recent trends in the selected characteristics. This averaging method not only consolidates the data, but it also captures short-term trends that are important for making accurate forecasts.

The calculated averages are then matched chronologically with the respective closing prices, establishing a smooth link between the averaged attributes and the goal variable—the current day's closing price.

To ensure consistency and fairness in the model training process, the averaged feature values are normalised using the well-known min-max scaling approach. This normalisation ensures that the characteristics remain on a consistent scale, avoiding any one feature from disproportionately impacting the learning process owing to differences in size.

Finally, the generated data is reshaped to comply to the unique needs of the selected neural network design, such as a 1D convolutional layer. This reshaping guarantees that the processed feature data is compatible with the neural network's expected input structure. Overall, this thorough data preparation process, which includes feature averaging, normalisation, and reshaping, places the dataset best for integration with CNN, RNN, or LSTM models, allowing for informed and dynamic forecasts of the current day's closing price.

### Final data predictions

The trained models forecast the new data, and the results are visualised using charts that compare projected values to actual values. Visualisation assists in interpreting model performance and reveals possible areas for improvement.

# Analysis

The analysis section of this project aims to convey the logical and conceptual components of the stock price prediction system. This section offers a comprehensive grasp of the overall architecture, the data representation attained for the system and its structure.

## System overview:

The system architecture outlines the high-level structure, beginning with the introduction of both the datasets used from external sources mainly being the historical stock price details from Yahoo Finance and the news articles from Seeking Alpha. The next part of the system traverses through the different stages of preprocessing, normalization, modeling, evaluation, and critical appraisal phases.

## Data Model:

The data model emphasizes logical representations of key entities and their relationships. The Financial Data Entity, Sentiment Analysis Entity, Merged Dataset Entity, Normalized Dataset Entity, Model Output Entity, and Evaluation Metrics Entity collectively form a comprehensive data structure.

This model elucidates the flow of information and the relationships among various data components.

1.**Financial Data Entity**:

The historical stock price data contains fields which are 'Open,' 'High,' 'Low,' 'Close,' 'Volume,' and 'Date.'

2.**Sentiment Analysis Entity**:

The sentiment data consists of the news article Titles and Dates for each entry from Seeking Alpha which is a website containing the required information and the Sentiments obtained after applying sentiment analysis using VADER.

3.**Merged Dataset Entity**:

The final dataset obtained after merging the historical stock prices and the sentiment dataset based on the common field which is the date for each data entry.

4.**Normalized Dataset Entity**:

This entity shows the dataset that has been normalized through post-Min-Max and removing missing values from the data.

5.**Model Output Entity**:

This entity represents the output generated by each model which mainly focuses on the predicted stock prices.

6.**Performance Evaluation Metrics Entity**:

The performance evaluations have been completed using metrics such as percentage accuracy, RMSE and R2 used for model evaluation.

By taking readers through the system's conceptual foundation, user interactions, and underlying data structures, this integrated study offers a thorough and coherent perspective. It provides a basis of knowledge for the next parts, which explore implementation details and interpretations of the results.

# Design

## Feature Scaling

In the project code, feature scaling is implemented using Min-Max scaling method. This is done to normalize the 'Close' price values in the main dataset. The Min-Max scaling method scales the values to a specific range, typically between 0 and 1.

## Convolutional Neural Networks (CNN)

Convolutional Neural Networks are a sort of neural network that is used to handle data that has a known grid-Along the lines of structure, such as time-series data (1-D grid of samples at regular time intervals) or picture data (2-D grid of pixels). The network employs a mathematical technique known as convolution rather than ordinary matrix multiplication in at least one of its layers, it is known as a convolutional neural network.

Lecun presented CNN as a network model in 1998. CNN is a type of feedforward neural network that excels at image processing and natural language processing. It may be used efficiently for time series forecasting. CNN's local perception and weight sharing may significantly reduce the number of parameters, enhancing model learning efficiency. CNN is made up of two layers: convolution layer and pooling layer as shown in figure 4 below.

A diagram of a process

Description automatically generated

Figure 4 : CNN architecture block diagram

Each convolution layer has several convolution kernels, and the calculation formula for each is provided in formula.

The features of the data are extracted after the convolution operation of the convolution layer, but the extracted feature dimensions are very large, so to solve this problem and reduce the cost of training the network, a pooling layer is added after the convolution layer to reduce the feature dimension:

where lt is the output value after convolution, tanh is the activation function, xt is the input vector, kt is the convolution kernel's weight, and bt is the convolution kernel's bias.

## Recurrent Neural Networks (RNN)

A family of neural networks known as recurrent neural networks (RNNs) is distinguished by the existence of cyclic connections between computational units. RNNs are superior to feed-forward networks in that they can handle any sequence of inputs because of their internal memory systems. An RNN's computational units are each equipped with an adjustable weight set and a time-varying real-valued activation. This results in a structure where the same set of weights is applied repeatedly across the network graph. Many RNNs use a set of parameters to find the values of their hidden units.

RNNs express their calculation in terms of switching between states, making sure the trained model always has the same amount of input. Every time step, the same transition function with the same parameters is used to maintain this consistency.

The reason for circumstance of this fundamental feature, RNNs stand out from other neural network architectures and are especially useful for applications involving sequential input.

Even though they are useful, simple RNN models have a limited internal structure with only one repeating module, which is usually represented by a single tanh layer. More complex designs, like as Long Short-Term Memory networks (LSTMs), were developed in response to the inherent problems with basic RNNs, such as learning long-term dependencies.

While both RNNs and LSTMs share the fundamental concept of recurrent connections, LSTMs outperform basic RNNs in tasks requiring the modelling of complex dependencies within sequential data due to their advanced architecture and ability to selectively retain or discard information over extended sequences. LSTMs are a variant of RNNs that address the shortcomings of basic RNNs by incorporating a more complex structure consisting of multiple interacting gates. Unlike basic RNNs, which feature a single tanh layer, LSTMs feature specialised memory cells and three gates—input, forget, and output—that allow them to selectively retain or discard information over prolonged sequences.

## Long Short-Term Memory Networks (LSTM)

LSTM is a special kind of RNN architecture, introduced in 1997 by Hochreiter and Schmidhuber. The LSTM network model was created to address the long-standing issues of gradient expansion and gradient disappearance in RNN. The reason for circumstance it has its own memory and can do pretty accurate forecasts, it has been frequently utilised in speech recognition, emotional analysis, and text analysis. In recent years, it has also been used in stock market predictions. A conventional RNN has only one repeating module and a basic internal structure. Four of the LSTM modules, on the other hand, are identical to normal RNN modules and work in a unique interactive mode.

The LSTM memory cell consists of three parts: the forget gate, the input gate and the output gate as shown in Figure 5.

A diagram of a computer network

Description automatically generated

Figure 5 : Long Short-Term Memory neural network Block Diagram

The research uses Long Short-Term Memory (LSTM) networks, which are a deep and recurrent form of neural networks, as seen in the picture. Recurrent networks are different from conventional feed-forward networks in that neurons in them can transmit data to a layer that is either preceding or identical, rather than just in one way. In that instance, data flow is not unidirectional, and the practical consequence is the presence of short-term memory in addition to the long-term memory that neural networks naturally acquire because of training.

By addressing the vanishing gradient problem that recurrent networks would encounter while handling lengthy data sequences, LSTM models sought to improve performance.

In figure 5 shown above, the different gates in a model using LSTM architecture are shown.

It does this by maintaining a continual error flow using unique components known as "gates," which enable weight modifications and the truncation of the gradient when its information is no longer required.

# Implementation

## Programming language and platform

The programming language used for implementation of the code is attained using Python language. The platform used to run this code is through Jupyter Notebook.

External Libraries used:

1.**Pandas**:

Pandas is a Python library for manipulating data collections. It includes tools for data analysis, cleansing, exploration, and manipulation. The name "Pandas" refers to both "Panel Data" and "Python Data Analysis" and was established in 2008 by Wes McKinney.

Pandas is the foundation for effective data manipulation and preparation. The system conveniently manages financial statistics by using its diverse data structures and operations. Pandas offers activities Along the lines of data cleansing, formatting, and the seamless organisation of time-series data, all of which are critical for accurate stock price forecasts.

2.**VADER** (**Valence Aware Dictionary and sEntiment Reasoner**):

Instead of TextBlob, VADER is used for sentiment analysis the particular reason for circumstance it is especially designed for analysing sentiment in textual data. VADER has a pre-built sentiment lexicon and excels at collecting subtle sentiment expressions, making it a strong option for analysing financial news sentiment and improving the overall stock price prediction model.

3.**TensorFlow and Keras**:

TensorFlow, in combination with Keras, works as the foundation for building and training neural network models. These libraries provide a dynamic and scalable environment for building Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. TensorFlow's flexibility and Keras's high-level abstraction work together to enable the building of accurate and adaptive stock price prediction models.

This implementation covers the whole development lifecycle, including data preparation, sentiment analysis with VADER, the creation and training of the required neural network models. This enables the system to not just capture detailed patterns within financial data, but also to adapt to the stock market's varying trends.

## Approach

Iterative approach is used in the implementation of the code for this project. The development process began with the creation and testing of a model that includes CNN, RNN, and LSTM architectures. The viability and possible limitations of combining sentiment analysis with historical stock data for prediction was also considered using this approach.

## Data preprocessing:

The preprocessing stage is a critical phase in stock price prediction, involving quiet few processes to enhance the quality of data.

Initially, data cleaning addresses inconsistencies and duplicates in the dataset by removing duplicate data and removing rows with null values in the dataset.

Normalization ensures consistent scaling, preventing dominance of specific features during model training. The closing price column which is the target column in the given data has been normalized using **min-max** method.

Data splitting into training and testing sets is an important step for model creation.

Utilizing the **VADER** tool, sentiment analysis scores are derived from news article titles and encoded for integration into machine learning models. This streamlined preprocessing ensures data readiness and reliability for subsequent stages in the project.

## Model Training and Evaluation:

Neural network models (CNN, RNN, LSTM) are trained using historical data and their performance is measured using metrics such as Accuracy and Mean Squared Error (MSE). The different performance metrics used are from the **Sklearn** prebuilt library and functions are obtained from the **metrics** section of the library.

## Visualization:

The system incorporates visualisation components to help in understanding model performance. This is completed using the ‘pyplot’ function in matplotlib library.

The actual vs predicted value closing price plots provides a qualitative assessment of the models forecasting skills. The plots between the predicted values and actual values for all three models have been developed. The analysis of these plots shows how accurate the prediction of the different models used are.

# Results

The stock price prediction code is evaluated using a thorough examination of essential metrics for each model mainly using Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM).

## Performance Evaluations

The performance evaluation is attained using metrics like Accuracy, Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and R2 (Coefficient of Determination) (R2) are among these measurements. The acquired findings are summarised in the table below:

Table 1 : Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **MSE** | **RMSE** | **R2** (Coefficient of Determination) |
| CNN | 96.77% | 7.85e-05 | 0.00886 | 0.99896 |
| RNN | 93.61% | 0.0003076 | 0.01754 | 0.99591 |
| LSTM | 95.79% | 0.0001333 | 0.01155 | 0.99823 |

## Result Analysis

The CNN model outperforms the others, with an accuracy of 96.77% and a remarkably low MSE of 7.85e-05. As indicated by its low RMSE of 0.00886, this model excels at capturing spatial patterns within financial data. The CNN model's ability to explain over 99.9% of the variance in closing prices is further demonstrated by the R2 value of 0.99896.

While the RNN model behind the CNN in accuracy, it still achieves a remarkable 93.61% accuracy and a competitive MSE of 0.0003076. The model's ability in capturing sequential dependencies is evident in its RMSE of 0.01754 and R2 value of 0.99591, suggesting excellent explanatory power.

With an accuracy of 95.79%, the LSTM model strikes a compromise between the CNN and RNN models. Its low MSE of 0.0001333 and RMSE of 0.01155 demonstrate its ability to capture long-term relationships. The R2 score of 0.99823 indicates a strong explanatory capability, which corresponds to the system's predicted accuracy.

Finally, the comprehensive analysis highlights each model's distinct capabilities, with the CNN model excelling in spatial pattern recognition, the RNN model capturing sequential dependencies and the LSTM model efficiently addressing long-term dependencies. These findings give important insights into the system's predictive capabilities, paving the way for further refinement and upgrades.

The below figures show the predicted values and actual values plots of all three models used.

The below figure 6. shows the plot for CNN Model with the actual and predicted values. CNN is the most accurate among the three models used.

A graph showing a graph of a graph

Description automatically generated with medium confidence

Figure 6 : Actual vs Predicted values - CNN.

The below figure 7. shows the plot for RNN Model with the actual and predicted values. RNN is the least accurate among the three models used.

A graph showing a graph

Description automatically generated with medium confidence

Figure 7 : Actual vs Predicted values - RNN.

The below figure 8. shows the plot for LSTM Model with the actual and predicted values. LSTM is moderately accurate among the three models used.

A graph of orange and blue lines

Description automatically generated

Figure 8 : Actual vs Predicted values - LSTM.

# Discussion

The literature review offers an informative overview of the field of stock price prediction, highlighting the dynamic nature of the financial market and the need of considering several factors that can affect the market.

The paper emphasises how deep learning models like LSTM, CNN and RNN architectures are becoming more and more popular for use in stock price predictions. The consideration of Sentiment Analysis of news articles also is a main component of this study.

The results section shows the effective use of deep learning models for stock price prediction. All models used seems to accurately predict the closing stock price values with the CNN model being the model with the highest accuracy.

The literature review of past studies showed that LSTM model provided the highest accuracy among the models used but the project in hand shows that CNN is the most accurate for the data used here. The sentiment-encoded feature using VADER sentiment analysis, improves the model’s predictive power even further.

One important point that is emphasised in the results and research is data normalisation. Our findings support the literature reviews assertion that normalising data increases model accuracy. The addition of sentiment analysis using VADER sentiment analysis to conventional techniques is a step up that complies with the literature's recommendation that sentiment and emotional factors be considered for more accurate stock price prediction. External factors are hence taken into consideration using sentiment analysis.

The results obtained, which use several models for prediction, are consistent with the literature review's investigation of combination models, such as CNN-LSTM. Our results are consistent with the literature suggesting that mixing designs might improve performance and accuracy.

The research highlights several drawbacks, for models not using sentiment analysis. In response, we recognise the significance of external factors in financial markets and want to overcome this constraint in future research to assure more accurate stock forecasts. The use of sentiment analysis implementation makes the selection of VADER over Text Blob and NLTK, as mentioned in the literature. In line with the conclusions of the research, VADER's effectiveness and awareness of contextual elements, such as emoticons and slang, add to its better performance.

In conclusion, the comparison of the literature review and the findings emphasises the effective use of sentiment analysis, deep learning model usage and normalisation approach consideration. The results and literature alignment highlights how crucial it is to combine sentiment analysis and sophisticated algorithms for reliable stock price prediction in this dynamic financial market.

# Project Management

Project management is essential for the effective completion of this project, as it ensures that objectives and goals for the research question are completed within set schedules and without compromising on the quality of the project.

This section focuses on three critical areas of project management: project scheduling, risk management, and quality assurance.

## Project Schedule

The project timeline was developed to cover all stages of the code development. A Gantt chart has been used to visualise project timeframes allowing for effective monitoring and tracking of objective deadlines.

Adjustments were made as needed to accommodate unanticipated obstacles while remaining on track with project objectives. The timetable was reviewed on a regular basis, allowing for adaptation with respect to changing project dynamics.

A graph with red lines

Description automatically generated

Figure 9 : Gantt chart for project timeline

## Risk Management

This project uses a thorough risk management approach to handle any possible problems, with an importance given to reducing the risks related to sentiment analysis and stock price prediction. Rigid data preparation techniques like removing missing values, normalisation and validation processes, efficiently reduce data-related hazards, such as biases and inconsistencies inside the sentiment analysis dataset. To maintain good data quality and protect project outcomes, routine validation checks are carried out to find and fix any anomalies.

Model overfitting is a significant risk factor that is frequently encountered in machine learning initiatives. To reduce this risk, strategies like cross-validation and regularisation are used.

Regular model validation activities are carried out to evaluate the sentiment analysis models' generalisation performance and confirm that they can handle newly discovered data without reducing the precision of stock price forecasts.

There are inherent risks associated with the accuracy of sentiment-based stock price estimates, including changes in market sentiment or unexpected economic developments. The project team uses flexibility in the models to successfully address these risks. The models' ability to adjust to changing market circumstances and get real-time updates makes them more resistant to outside influences.

Throughout the project, risk identification is a continuous process that involves frequent team meetings and evaluations. To guarantee the dependability and robustness of sentiment analysis models in the context of stock price prediction, the project team proactively detects and resolves such problems. This thorough approach to risk management not only helps the project succeed overall but also improves its capacity to handle the uncertainties present in data-driven research and financial markets.

## Quality Management

Quality management was critical in assuring the dependability and correctness of the study findings. Throughout the project's lifespan, industry standards and best practises were followed. To analyse progress and refine outcomes, techniques such as code reviews, model validation, and continuous performance monitoring were used.

Regular quality checks were performed to ensure that project outputs fulfilled set standards. This iterative approach to quality management allows for continual development, resulting in strong and dependable final models. The dedication to quality standards contributed to the study findings legitimacy and validity.

## Social, Legal, Ethical and Professional Considerations

The Social, Legal, Ethical and Professional problems have been considered by carefully investigating the societal implications of predictive modelling, including its possible influence on financial markets, to guarantee ethical and transparent research processes.

Legal and ethical frameworks served as the direction for approach, supporting the project's adherence to privacy laws, ethical standards, data protection and professional behaviour guidelines. The project's societal importance and ethical integrity has been improved by ensuring that sensitive financial data was handled properly.

# Critical Appraisal

The critical analysis of this study project comprises a comprehensive and unbiased assessment of its many aspects, including both areas designated for improvement and those that are praiseworthy. The information and experience gathered throughout the course of the project are summarised in this section of the report.

## Positive Analysis:

Combining the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models demonstrates a sophisticated investigation and offers insightful information about their relative effectiveness. The project's resilience is strengthened by this variety, which puts it at the forefront of current financial predictive modelling research.

One major accomplishment is the sentiment analysis's effective integration. Incorporating qualitative market sentiment data into the quantitative domain gives the prediction models additional nuance. This conformity to current trends in financial predictive modelling highlights the flexibility and insight of the project in identifying the mutually beneficial interaction between qualitative and quantitative data.

Strict project management guidelines have served as the foundation for this undertaking. The project's openness and trustworthiness are enhanced by the thorough documentation of scheduling, risk assessment, and ethical issues. This dedication to meticulous project management demonstrates the professionalism displayed throughout the project's lifespan.

Moreover, the project's practical relevance is apparent due to its emphasis on financial market stock price forecasting. Observing the models' forecasts materialise within the framework of real market movements offers concrete proof of the project's applicability and potential effect.

## Opportunities for Development:

The initiative recognises its strengths but also points out important areas that need to be improved. It is critical to address data constraints, and major tasks include improving data quality, investigating larger datasets, and putting bias management techniques into practise. These improvements would fill in any information gaps and strengthen the training and generalisation of the models.

Improving the interpretability of models is considered crucial, particularly for intricate architectures along the lines of CNN and LSTM. The inability to comprehend in depth is a problem for practical implementation. Subsequent versions should investigate approaches that offer a more detailed comprehension of the reasoning behind certain forecasts, promoting increased confidence in the models decision-making procedures.

Overfitting problems are still being addressed. It is necessary to investigate methods further to improve the generalisation and resilience of the model to varying market circumstances. Techniques to increase the model’s flexibility to unknown data would increase their efficacy in a variety of market situations.

The approach emphasises ongoing ethical evaluation when it enters delicate areas along the lines of banking. Ensuring the project maintains ethical integrity throughout its lifespan is deemed vital, requiring ongoing participation with ethical conversations and the monitoring of any ramifications.

## Acquired Knowledge and Proficiency:

The initiative has provided a means of acquiring priceless information and experience. One important result has been the mastery of deep learning techniques, such as CNN, RNN and LSTM models. A deeper comprehension of these complex designs' practical uses has resulted from navigating their complexities.

Sentiment analysis has been successfully integrated, demonstrating a sophisticated grasp of natural language processing and its consequences for financial forecasts. This knowledge puts the study at the forefront of studies examining the mutually beneficial link between qualitative and quantitative data, which represents a major advancement in multidisciplinary research.

A greater understanding of the moral obligations inherent in research has resulted from navigating ethical issues in the context of financial data and sentiment analysis. This information highlights the significance of ethical issues in data-driven research and is necessary for responsible and ethical research practises.

Adopting an iterative research approach that incorporates input, addresses obstacles, and refines procedures shows a thorough awareness of the dynamic nature of research. This iterative approach promotes adaptation and continual progress, making it a useful tool for future research projects.

## In conclusion:

Finally, the critical evaluation points out the benefits and drawbacks of the research completed. The study also promotes the use of technical and ethical knowledge and skills that will make a valuable contribution to the field of financial markets predictive modelling. This research study lays the groundwork for further investigations and continuous improvement in the field of data-driven research.

# Conclusions

This research project aimed at predicting stock prices through the integration of sentiment analysis and advanced deep learning models(CNN,RNN and LSTM) revealing nuanced findings and results across various variables.

The comprehensive exploration, blending Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) models, in combination with sentiment analysis of news articles related to the particular stock using VADER sentiment analysis has proved to accurately predict the closing prices of the stock for the current day.

## Deep learning model performance

This research project's main objective is on comparing and evaluating three different deep learning models for stock price prediction: long short-term memory (LSTM), recurrent neural network (RNN) and convolutional neural network (CNN).

The CNN model is the majority of accurate in predicting stock values among them, making it stand out as the top performer. This precision demonstrates the model's capacity to identify complex spatial patterns in financial data, which is very important when it comes to sentiment analysis.

A major factor in the CNN architecture's excellent predicting skills is how well it captures intricate correlations and patterns in the selected features and sentiment data. Convolutional layers in the model efficiently pick up hierarchical feature representations, which enables it to recognise and apply pertinent data from the historical prices and sentiment analysis dataset.

Consequently, the CNN model exhibits an unparalleled degree of accuracy, so demonstrating its promise as a resilient instrument for financial forecasting.

Although the CNN model is the front-runner, the separate contributions of the RNN and LSTM models must also be recognised. Designed to capture sequential relationships in data, the RNN model performs admirably with a somewhat lower accuracy than the CNN. Additionally performing well and finding a compromise between the CNN and RNN is the LSTM, which is renowned for managing long-term dependencies.

Essentially, the predictive model performance emphasises the unique qualities of each deep learning architecture in addition to highlighting the importance of sentiment analysis. This detailed analysis lays the groundwork for future refinement and optimization, providing valuable insights for researchers in stock price prediction projects.

## Importance of Sentiment analysis

An important part of this research endeavour is sentiment analysis, which turns out to be a major advantage in terms of improving the deep learning models capacity for stock price prediction. By adding a layer of contextual information gained from news articles pertained to that stock, sentiment analysis is included into the models, giving them a dynamic and real-time input.

The thorough data pretreatment procedures highlight the importance of sentiment analysis. To guarantee its quality and dependability, the sentiment data goes through extensive cleaning, normalisation and validation processes. The basis for the succeeding modelling processes is laid at this critical preliminary stage and biases or inconsistencies in the sentiment data might affect the precision of predictions.

Sentiment analysis plays a major role in helping to validate the models. Strict validation procedures are carried out, with an emphasis on determining how well the sentiment data corresponds with actual market movements. To make sure that the sentiment analysis models accurately reflect and capture the sentiments of market players, this validation stage is crucial.

Sentiment analysis has a significant role in improving the models' capacity to adjust to outside influences. News, social media, and world events all have an intrinsic impact on financial markets. The sentiment analysis for this project incorporates the sentiments of news articles. Sentiment analysis gives the models the capacity to take these outside factors into account, which enables them to adjust and improve forecasts in reaction to shifting market conditions.

The algorithms can identify not only past price movements but also the general sentiment of the market by adding sentiment analysis. This more complex knowledge helps to increase the accuracy of predictions, particularly in situations when news, emotion, or social media conversations impact market movements.

To sum up, sentiment analysis is very important to the project since it improves data quality, helps validate the models and makes the models more flexible in response to outside influences. The deliberate integration of sentiment analysis is credited in part for the project's success in stock price prediction, since it plays a crucial role in using non-numeric data to improve the deep learning models' overall forecasting ability.

## Drawbacks

Despite the project's overall success, it is important to acknowledge certain drawbacks and limitations that may influence the interpretation and generalization of the findings obtained. These limitations provide valuable insights for future research endeavours and highlight areas for improvement:

A major drawback is related to data availability and quality, which are essential components of financial predictive modelling. The dependability of past stock and sentiment data affects how accurate stock price forecasts may be. Inconsistencies in the data, potential biases, and information shortages might hinder the models' ability to learn and make predictions. In the future, research projects should focus on improving data quality, investigating larger datasets, and developing strategies to deal with incomplete or missing data.

Furthermore, due to their intrinsic dynamic nature, financial markets are susceptible to unanticipated occurrences that might drastically depart from past trends. It may be difficult for the models created for this research to correctly forecast abrupt occurrences or extremely volatile market movements. Given the intrinsic complexity of market dynamics, more research and the creation of cutting-edge methods may be necessary to refine models to handle such unanticipated situations.

Throughout the modelling process, the project makes a few assumptions and simplifications. For example, it assumes that collected sentiments and stock price fluctuations have a direct and linear relationship. On the other hand, the complex interaction of several elements in financial markets suggests a more subtle relationship between emotion and stock prices. To fully capture the complex dynamics of market sentiment, future research might explore more advanced sentiment analysis approaches.

Moreover, an issue with deep learning models is their interpretability, especially with sophisticated architectures Along the lines of CNN, RNN, and LSTM. The models' openness is limited since it is still difficult to understand the reasoning behind some of the forecasts. Future research should focus on improving model interpretability since it increases users' comprehension and confidence in these sophisticated models' decision-making processes.

Despite efforts to reduce its impacts, overfitting—a major problem in machine learning—remains a factor to be considered. When presented with fresh, untrained data, models that perform remarkably well on training data may find it difficult. Subsequent investigations must delve into methods for augmenting model generalisation and efficacy on a variety of datasets, guaranteeing their flexibility in response to changing market circumstances.

Even if they are acknowledged and somewhat handled, ethical issues are still a problem. Constant observation is required due to ethical issues surrounding the application of sentiment research in financial markets, such as the possibility of market manipulation using sentiment-driven tactics. To guarantee the appropriate development and deployment of predictive models in financial contexts, it is imperative that the research community engages in ongoing ethical assessments and debates.

To sum up, the shortcomings and restrictions found in this study provide important benchmarks for further investigation and improvement. By tackling these issues, we can build more reliable and morally acceptable financial market prediction models that keep up with the field's constant evolution.

# Student Reflections

Upon completing this project, I have gained practical insights into the complex and subtleties of predictive modelling within the financial markets. It has been a truly transformational experience. The comments I have as a student on this project include a wide range of topics, including the difficulties encountered as well as the valuable experiences I have learned:

## Implementation in Deep Learning Models:

A thorough examination of cutting-edge deep learning architectures, such as Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN), was required for this research. My understanding of these models' practical applicability has been much improved by navigating through their intricate details. Through this implementation I can now easily use and apply deep learning models into any research sorted after.

## Sentiment analysis integration:

An important turning point in the study was when sentiment analysis was included to the prediction models. It expanded my knowledge of natural language processing and made me see how important non-numeric data is for financial forecasting. Comprehending the influence of sentiment on market dynamics has enhanced my analytical toolbox with a new level of complexity.

## Difficulties in Preprocessing Data:

The project demonstrated how important data preparation is to predictive model success. Managing normalisation, validation, and quality control concerns provide practical experience handling real-world data problems. Without a doubt, these teachings will come in handy for next tasks and career endeavours.

## Iterative approach:

The project's iterative methodology reflected the reality of research. The experiment demonstrated the iterative and dynamic character of research, from overcoming constraints through continual improvement to fine-tuning models in response to input. This iterative method will serve as a foundation for how we approach our next research projects.

## Real World Data implementation:

The project's practical applicability, particularly in the financial arena, established a concrete link between theoretical knowledge and real-world applications. Observing the models forecasts in relation to real stock price fluctuations highlighted the influence that well-crafted models may have on decision-making procedures.

## Capabilities for Project Management and Ethical considerations:

My project management abilities were refined by managing a project of this magnitude. The project made me aware of the many facets involved in managing intricate research projects, from careful planning and risk reduction to ensuring no compromise in data quality. The procedure made it clear how important it is to follow deadlines and modify plans in the event of unanticipated difficulties.

Throughout the research, ethical issues were paramount, particularly when pertaining to sentiment analysis and financial data. It was underlined how important it is to exercise responsibility when working in delicate fields by navigating the ethical terrain of data management and model creation. As a researcher, these factors have surely influenced my ethical compass.

## Supervisor Meetings and feedback

The meetings with my supervisor transcended traditional mentorship, evolving into a collaborative learning experience. The exchange of ideas, critical feedback and insights fostered an environment where both expertise and curiosity were valued.

Not only has the sequence of meetings with my supervisor shaped the technical components of the study, but it has also helped me grow as a student researcher. These encounters' collaborative, iterative, and guided style has deeply impacted my research methodology and will surely have an impact on my future academic and non-academic endeavours.

The completion of this assignment is only a first step towards further education. Understanding the shortcomings and possible areas for development has sparked an interest in learning more about improving models, investigating other approaches, and keeping up with developments in the always changing field of predictive modelling.

All things considered, this project has been a lively and stimulating experience that provides a comprehensive understanding of the research process, from conception to execution. The knowledge gained, obstacles surmounted, and moral dilemmas resolved will surely influence my future work as a scholar and prospective researcher. The project's success is not only determined by the results attained but also by the long-lasting influence it has had on my development and comprehension of the field of inquiry.

Bibliography and References

1. Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. *Complexity*, *2020*, 1-10.

2. Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, *15*, 2352.

3. Bonta, V., Kumaresh, N., & Janardhan, N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1-6.

4. *Stock market’s price movement prediction with LSTM neural networks*. (2017, May 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/7966019>

5. *A Deep Multimodal Reinforcement Learning System Combined with CNN and LSTM for Stock Trading*. (2019, October 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/8939991>

6. *Prediction of Stock Market Using Recurrent Neural Network*. (2021, October 27). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9623206>

7. Pramoditha, R. (2022, June 29). *Convolutional Neural Network (CNN) Architecture Explained in Plain English Using Simple Diagrams*. Medium. <https://towardsdatascience.com/convolutional-neural-network-cnn-architecture-explained-in-plain-english-using-simple-diagrams-e5de17eacc8f>

8. *vaderSentiment*. (2020, May 22). PyPI. <https://pypi.org/project/vaderSentiment/>

9. *Yahoo is part of the Yahoo family of brands*. (n.d.). <https://finance.yahoo.com/quote/TSLA.MI/history?p=TSLA.MI>

10. *Tesla, Inc. (TSLA) Latest Stock News*. (n.d.). Seeking Alpha. <https://seekingalpha.com/symbol/TSLA/news>

11. Patro, S. G. K. (2015, March 19). *Normalization: A Preprocessing Stage*. arXiv.org. <https://doi.org/10.48550/arXiv.1503.06462>

12. Moghar, A., & Hamiche, M. (2020, January 1). *Stock Market Prediction Using LSTM Recurrent Neural Network*. Procedia Computer Science. <https://doi.org/10.1016/j.procs.2020.03.049>

13. Prabowo, R., & Thelwall, M. (2009, April 1). *Sentiment analysis: A combined approach*. Journal of Informetrics. <https://doi.org/10.1016/j.joi.2009.01.003>

Appendix A – Interim Progress Report and Meeting Records

Meeting Details and updates

|  |  |  |  |
| --- | --- | --- | --- |
| **SI No** | **Date** | **Project Supervisor** | **Project Updates** |
| 1. | 28-10-23 | James Brusey | First meeting to discuss project requirements and goals. |
| 2. | 17-10-23 | Beate Grawemeyer | Project topic selection, Dataset selection |
| 3. | 24-10-23 | Beate Grawemeyer | Objective finalization, Ethics approval |
| 4. | 31-10-23 | Beate Grawemeyer | Implementation of code, Implementation of objectives and report , first draft preparation |
| 5. | 07-11-23 | Beate Grawemeyer | Development of Python Modules |
| 6. | 14-11-23 | Beate Grawemeyer | Integration of machine Learning Algorithms |
| 7. | 21-11-23 | Beate Grawemeyer | Testing And Debugging |
| 8. | 28-11-23 | Beate Grawemeyer | Deployment and User Training |
| 9. | 05-12-23 | Beate Grawemeyer | Documentation and Finalisation |

Appendix B – Certificate of Ethics Approval

A certificate of ethical approval

Description automatically generated

Appendix X – Python Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266 | **import** **pandas** **as** **pd**  **import** **requests**  **import** **re**  **from** **bs4** **import** BeautifulSoup  **from** **textblob** **import** TextBlob  **from** **datetime** **import** datetime, timedelta  **from** **nltk.sentiment** **import** SentimentIntensityAnalyzer  **from** **sklearn.metrics** **import** mean\_squared\_error, r2\_score  **from** **sklearn.model\_selection** **import** train\_test\_split  **from** **sklearn.preprocessing** **import** StandardScaler  **import** **pandas** **as** **pd**  **import** **numpy** **as** **np**  **from** **sklearn.model\_selection** **import** train\_test\_split  **from** **sklearn.linear\_model** **import** LinearRegression  **from** **sklearn.metrics** **import** mean\_squared\_error  **import** **matplotlib.pyplot** **as** **plt**  **from** **sklearn.preprocessing** **import** StandardScaler  **from** **keras.models** **import** Sequential  **from** **keras.layers** **import** Conv1D,MaxPooling1D,Flatten,Dense,LSTM,SimpleRNN  **import** **nltk**  nltk.download('vader\_lexicon')  dataset = pd.read\_csv(r'C:\Users\karth\Downloads\TSLA (3).csv')  dataset.head()  #Extracting News article titles  **def** **format\_date**(date\_text):  **if** "Yesterday" **in** date\_text:  **return** (datetime.now() - timedelta(days=**1**)).strftime("%d-%m-%Y")  **elif** "Today" **in** date\_text:  **return** datetime.now().strftime("%d-%m-%Y")  **elif** re.match(r'^[A-Za-z]+, [A-Za-z]+\. \d{1,2}, \d{4}$', date\_text):  **return** datetime.strptime(date\_text, "%a, %b. %d, %Y").strftime("%d-%m-%Y")  **elif** re.match(r'^[A-Za-z]+, [A-Za-z]+ \d{1,2}, \d{4}$', date\_text):  **return** datetime.strptime(date\_text, "%a, %b %d, %Y").strftime("%d-%m-%Y")  **elif** re.match(r'^[A-Za-z]+, [A-Za-z]+ \d{1,2}, \d{4}$', date\_text):  **return** datetime.strptime(date\_text, "%a, %b %d, %Y").strftime("%d-%m-%Y")  **elif** re.match(r'^[A-Za-z]+, [A-Za-z]+ \d{1,2}$', date\_text):  date\_text = date\_text + ", 2023"  **return** datetime.strptime(date\_text, "%a, %b %d, %Y").strftime("%d-%m-%Y")  **elif** re.match(r'^[A-Za-z]+, [A-Za-z]+\. \d{1,2}$', date\_text):  date\_text = date\_text + ", 2023"  **return** datetime.strptime(date\_text, "%a, %b. %d, %Y").strftime("%d-%m-%Y")  **else**:  **print**("Unrecognized date format:", date\_text)  **return** None  pg\_no = **130** #Extracting news articles from webpage having 130 pages  data = {'Title': [], 'Date': [], 'Sentiment Analysis': []}  #Applying sentiment analysis using VADER on extracted news titles  sid = SentimentIntensityAnalyzer()  **print**("Extracting news article titles from webpage : https://seekingalpha.com/symbol/TSLA/news")  **for** i **in** range(**1**, pg\_no + **1**):  url\_base = "https://seekingalpha.com/symbol/TSLA/news?page="  url = url\_base + str(i)  response = requests.get(url)  **if** response.status\_code == **200**:  soup = BeautifulSoup(response.text, 'html.parser')  article\_info = soup.find\_all('div', class\_='grow')  **for** info **in** article\_info:  title\_elem = info.find('h3', class\_='text-share-text')  date\_elem = info.find('span', class\_='whitespace-nowrap sa-circle-divider-share-text-2')  **if** title\_elem **and** date\_elem:  title = title\_elem.find('a')  date\_text = date\_elem.text.strip()  formatted\_date = format\_date(date\_text)  **if** title.text.strip() **not** **in** data['Title']:  # Use VADER for sentiment analysis  polarity = sid.polarity\_scores(title.text.strip())['compound']  **if** polarity > **0**:  sentiment = 'Positive'  **elif** polarity < **0**:  sentiment = 'Negative'  **else**:  sentiment = 'Neutral'  data['Title'].append(title.text.strip())  data['Date'].append(formatted\_date)  data['Sentiment Analysis'].append(sentiment)    df = pd.DataFrame(data)  df\_set = pd.DataFrame(dataset)  #converting date to required format  df['Date'] = pd.to\_datetime(df['Date'], format='%d-%m-%Y')  df.head()  df\_set = df\_set.sort\_values(by='Date', ascending=False)  df\_set.head()  df['Date'] = pd.to\_datetime(df['Date'], format='%Y-%m-%d')  df\_set['Date'] = pd.to\_datetime(df\_set['Date'], format='%Y-%m-%d')  grouped\_sentiments = df.groupby('Date')['Sentiment Analysis'].agg(**lambda** x: x.mode().iloc[**0**]).reset\_index()  #merging datasets  merged\_df = pd.merge(df\_set,grouped\_sentiments, on='Date', how='left')  **print**(merged\_df)  merged\_df.head()  #preprocessing stage  **from** **sklearn.preprocessing** **import** MinMaxScaler  # Min-Max scaling  minmax = MinMaxScaler().fit(merged\_df.iloc[:, **4**:**5**].astype('float32'))  df\_log = minmax.transform(merged\_df.iloc[:, **4**:**5**].astype('float32'))  df\_log = pd.DataFrame(df\_log)  df\_log.head()  merged\_df['Scaled\_Close'] = df\_log  merged\_df.head()  sentiment\_mapping = {'Negative': **0**, 'Neutral': **1**, 'Positive': **2**}  merged\_df['Sentiment\_Encoded'] = merged\_df['Sentiment Analysis'].map(sentiment\_mapping)  merged\_df.drop('Sentiment Analysis', axis=**1**, inplace=True)  **print**(merged\_df.head())  merged\_df.dropna(inplace=True)  **print**(merged\_df)  #features = ['Open', 'High', 'Low', 'Volume','Sentiment\_Encoded']  #target = ['Scaled\_Close']  X = merged\_df[['Open', 'High', 'Low', 'Volume', 'Sentiment\_Encoded']].values  y = merged\_df['Scaled\_Close'].values  #Splitting data  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=**0.2**, random\_state=**42**)  scaler = StandardScaler()  X\_train1 = scaler.fit\_transform(X\_train)  X\_test1 = scaler.transform(X\_test)  #CNN Model implementation  X\_train = X\_train1.reshape(X\_train1.shape[**0**], X\_train1.shape[**1**], **1**)  X\_test = X\_test1.reshape(X\_test1.shape[**0**], X\_test1.shape[**1**], **1**)  model = Sequential()  model.add(Conv1D(filters=**64**, kernel\_size=**3**, activation='relu', input\_shape=(X\_train.shape[**1**], **1**)))  model.add(MaxPooling1D(pool\_size=**2**))  model.add(Flatten())  model.add(Dense(**64**, activation='relu'))  model.add(Dense(**1**, activation='linear'))  model.compile(optimizer='adam', loss='mean\_squared\_error')  model.fit(X\_train, y\_train, epochs=**10**, batch\_size=**32**, validation\_data=(X\_test, y\_test))  loss = model.evaluate(X\_test, y\_test)  **print**(f'Mean Squared Error on Test Data: {loss}')  predictions = model.predict(X\_test)  plt.figure(figsize=(**12**, **6**))  plt.plot(y\_test, label='Actual', linestyle='-', color='blue')  plt.plot(predictions, label='Predicted (CNN)', linestyle='-', color='orange')  plt.title('Actual vs Predicted Stock Prices using CNN')  plt.xlabel('Time')  plt.ylabel('Scaled Close Price')  plt.legend()  plt.grid(True)  plt.show()  #RNN model implementation  X\_train\_rnn = X\_train1.reshape(X\_train1.shape[**0**], **1**, X\_train1.shape[**1**])  X\_test\_rnn = X\_test1.reshape(X\_test1.shape[**0**], **1**, X\_test1.shape[**1**])  model\_rnn = Sequential()  model\_rnn.add(SimpleRNN(**50**, activation='relu', input\_shape=(X\_train\_rnn.shape[**1**], X\_train\_rnn.shape[**2**])))  model\_rnn.add(Dense(**1**))  model\_rnn.compile(optimizer='adam', loss='mean\_squared\_error')  model\_rnn.fit(X\_train\_rnn, y\_train, epochs=**10**, batch\_size=**32**, validation\_data=(X\_test\_rnn, y\_test))  predictions\_rnn = model\_rnn.predict(X\_test\_rnn)  plt.figure(figsize=(**12**, **6**))  plt.plot(y\_test, label='Actual', linestyle='-', color='blue')  plt.plot(predictions\_rnn, label='Predicted (RNN)', linestyle='-', color='orange')  plt.title('Actual vs Predicted Stock Prices using RNN')  plt.xlabel('Time')  plt.ylabel('Scaled Close Price')  plt.legend()  plt.grid(True)  plt.show()  #LSTM model implementation  X\_train\_lstm = X\_train1.reshape(X\_train1.shape[**0**], X\_train1.shape[**1**], **1**)  X\_test\_lstm = X\_test1.reshape(X\_test1.shape[**0**], X\_test1.shape[**1**], **1**)  model\_lstm = Sequential()  model\_lstm.add(LSTM(**50**, activation='relu', input\_shape=(X\_train\_lstm.shape[**1**], X\_train\_lstm.shape[**2**])))  model\_lstm.add(Dense(**1**))  model\_lstm.compile(optimizer='adam', loss='mean\_squared\_error')  model\_lstm.fit(X\_train\_lstm, y\_train, epochs=**10**, batch\_size=**32**, validation\_data=(X\_test\_lstm, y\_test))  predictions\_lstm = model\_lstm.predict(X\_test\_lstm)  plt.figure(figsize=(**12**, **6**))  plt.plot(y\_test, label='Actual', linestyle='-', color='blue')  plt.plot(predictions\_rnn, label='Predicted (LSTM)', linestyle='-', color='orange')  plt.title('Actual vs Predicted Stock Prices using LSTM')  plt.xlabel('Time')  plt.ylabel('Scaled Close Price')  plt.legend()  plt.grid(True)  plt.show()  #Performance evaluations  **def** **calculate\_and\_print\_accuracy**(y\_true, y\_pred, model\_name):  mse = mean\_squared\_error(y\_true, y\_pred)  rmse = np.sqrt(mse)  r2 = r2\_score(y\_true, y\_pred)  accuracy\_percentage = **100** \* (**1** - rmse / np.std(y\_true))  **print**('Performance metrics for:',model\_name)  **print**('Accuracy:',accuracy\_percentage)  **print**('Mean Squared Error (MSE):', mse)  **print**('Root Mean Squared Error (RMSE):', rmse)  **print**('R squared values(R2): ',r2)  **print**('**\n**')    calculate\_and\_print\_accuracy(y\_test, predictions, 'CNN')  calculate\_and\_print\_accuracy(y\_test, predictions\_rnn, 'RNN')  calculate\_and\_print\_accuracy(y\_test, predictions\_lstm, 'LSTM') |