**FinTalk Predictive AI Assistnant**

**Submitted for**

Statistical Machine Learning CSET211

Submitted by:

**K Sridhar Reddy (E23CSEU1817)**

Submitted to:

**Amit Soni Arya**

**July-Dec 2024**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

1. **Abstract**

This Project shows how we can use a special type of machine learning model, called LSTM (Long Short-Term Memory), to predict stock prices. By studying past stock data, the LSTM model learns patterns and trends, helping us make predictions about future prices. The notebook explains how we prepare the data, train the model, and check how accurate its predictions are. This project helps us understand if LSTM models can be useful for predicting stock market trends, which might be helpful for making investment decisions.

1. **Introduction**

In the world of finance, stock prices are influenced by various factors and often change unpredictably. This project aims to explore how machine learning can help predict these price movements by using historical stock data. We use a model called Long Short-Term Memory (LSTM), which is particularly suited for time-related data, as it can remember key information from past events. The project involves preparing and processing stock data, then training the LSTM model to recognize patterns and trends. While challenges exist in making highly accurate predictions, this project highlights the potential of machine learning for gaining insights into stock market behavior.

**Challenges:**

1. **Unpredictable Market**
2. **Data Issues**
3. **Overfitting**
4. **Choosing the Right Data**
5. **Complexity**

**Contributions:**

**1)Using LSTM for Stock Prediction**

**2)Improving Prediction Methods**

**3)Data Preparation**

**4)Model Performance Review**

**5)Opening Path for Future Work**

1. **Related Work :-**

In this Project, we are going to build a model using real-world data derived from some of our production systems. This data gives a very close picture of some of the things we have to do every day to be successful at trading in modern financial markets. We’ve assembled a collection of features and responders related to markets where we run automated trading strategies and are concerned about having good underlying models. To balance crafting a challenging, relevant problem that ties into our business while respecting the proprietary and highly competitive nature of our trading, you will notice that we have anonymized some of the features and responders we present in the data.

These models help us actively trade thousands of financial products each day across 200+ trading venues around the world. While this challenge only presents a tiny fraction of the quantitative problems work on daily.

In this project we will discover and explore data from the stock market, particularly some technology stocks (Apple, Amazon, Google, and Microsoft). We will learn how to use yfinance to get stock information, and visualize different aspects of it using Seaborn and Matplotlib. we will look at a few ways of analyzing the risk of a stock, based on its previous performance history. We will also be predicting future stock prices through a Long Short Term Memory (LSTM) method.

These related works highlight the use of various machine learning techniques, particularly LSTMs, in stock market prediction. Many studies focus on improving the accuracy of predictions by exploring hybrid models or combining data from multiple sources, such as sentiment analysis, to better understand stock price movements.

1. **Methodology**

1. **Data Pre-processing**

This step involves preparing your raw data for analysis. Since financial data is often messy or incomplete, this phase is critical.

**1)Handling Missing Data**

**2)Outlier Detection**

**3)Feature Engineering**

**4)Scaling Data**

2. **Data Visualization / EDA (Exploratory Data Analysis)**

EDA allows you to explore and visualize the dataset to uncover patterns, correlations, and trends, which is crucial before feeding the data into models. For both projects:

* **Price Trends Visualization**
* **Correlation Analysis**
* **Volatility Analysis**
* **Return Distributions**

3. **Model Creation and Testing**

Once data is pre-processed and explored, the next step is model building. For both projects, you’ll want to experiment with multiple machine learning and statistical models. The types of models you choose will depend on whether you’re forecasting trends, making classifications, or offering decision-making advice.

**Model 1: Random Forest (for Regression or Classification)**

Random Forest is a machine learning method used for predicting values (like stock prices) or classifying outcomes (like whether stock prices will go up or down). It avoids overfitting, handles complex data well, and is great for models that consider many factors like price and volume. It is tested by measuring errors for predictions or accuracy for classifications.

**Model 2: LSTM (Long Short-Term Memory)**

LSTM is a type of neural network that works well with time-based data because it remembers patterns over long periods, which is helpful for predicting financial trends where past prices affect future ones. Unlike older models, LSTMs can handle complex and long-term patterns, making them great for forecasting stock prices or market movements. To check how good an LSTM model is, we use error measurements like MSE or RMSE and test it on new data with train-test splits or cross-validation.

**Testing the Models**

Regardless of the model used, we should need to evaluate its performance rigorously:

**Backtesting**: A crucial step in financial forecasting where you run your model on historical data to see how it would have performed in past markets. Backtesting helps validate that your model can generalize across various market conditions.

**Real-Time Testing**: In a real-time scenario, especially for the **Real-Time Market Data Forecasting** project, you will evaluate how well your model reacts to live data. Models need to be retrained periodically to account for changes in market dynamics.

1. **Hardware/Software Required**

**Hardware Requirements:**

**Computer or Laptop**

**Software Requirements:**

**Operating System**: Windows or macOS

**Programming Language**: **Python** (version 3.6 or higher).

**Libraries/Tools**:

* 1. **TensorFlow** or **Keras**
  2. **Pandas**
  3. **NumPyMatplotlib/Seaborn**
  4. **Scikit-learn**

**Development Environment**:

* 1. **Jupyter Notebook**: Easy tool for coding and testing.
  2. Or, use **VS Code** or **PyCharm** if you prefer.

**Data Sources**:

* 1. **Yahoo Finance API** or **Alpha Vantage API**: For fetching stock data.
  2. Alternatively, you can use **CSV files** with stock data.

1. **Experimental Results :-**

**Training and Test Performance**:

* After training the LSTM model on historical stock data, we evaluated its performance on both the training and test datasets. The training loss, measured using Mean Squared Error, decreased steadily, showing that the model learned patterns from the historical data. On the test dataset, the model performed well, with predictions closely matching actual stock prices, though some fluctuations occurred due to market volatility.

· **Visualization of Predictions**:

* A line chart was created to compare **actual stock prices** with the **predicted prices** over a specific time period.
* The predicted prices followed the trend of actual prices, capturing general price movement patterns.
* Minor deviations between the actual and predicted values were noted, especially around rapid price changes, where the model struggled to predict sharp increases or decreases accurately.

**Evaluation Metrics**:

* **Mean Squared Error (MSE)** and **Root Mean Squared Error (RMSE)** were calculated to assess prediction accuracy.
* These metrics showed that while the model was not perfect, it was effective in capturing the general trend of the stock prices.
* Lower MSE and RMSE values indicated that the LSTM model provided reasonable accuracy in stock price prediction.

**Strengths and Limitations**:

* **Strengths**: The model successfully captured price trends and was able to predict smoother patterns.
* **Limitations**: The LSTM model struggled with sudden, unpredictable price spikes or drops, showing that improvements could be made for handling market volatility.

· **Future Improvements**:

* Adding more data, like trading volume or sentiment analysis data, may help improve accuracy.
* Experimenting with additional models or hybrid approaches could further refine the predictions.

**7 . Conclusions :-**

This project shows that LSTM models can help predict stock prices by learning from past data. The model successfully captured general price trends, proving that machine learning can be useful in analyzing stock market patterns. However, it had trouble with sudden, unexpected price changes, which are difficult to predict.

LSTM models are helpful for understanding stock trends but need more improvements to work well with real-world market ups and downs. Adding extra information, like trading volume or news sentiment, or combining LSTM with other methods could make the model more accurate and reliable for stock prediction.

**8 . GitHub Link of Complete Project**

https://github.com/Sridhar2233/Project