Report on Stock Prediction Model using Long Short-Term Memory (LSTM)

1.Introduction

Stock price prediction is a challenging task due to the non-linear, volatile, and time-dependent nature of financial data. Long Short-Term Memory (LSTM), a type of Recurrent Neural Network (RNN), is well-suited for time-series prediction because it can capture temporal dependencies and retain information over longer sequences.



2. Objective

The objective of this project is to design and evaluate an LSTM-based model for predicting the future prices of a selected stock. The model will be trained on historical stock price data and tested on unseen data to assess its predictive capabilities.



3. Data Collection and Preprocessing

3.1 Data Source

Historical stock price data is obtained from reliable sources, such as Yahoo Finance or Alpha Vantage API, including Open, High, Low, Close (OHLC) prices, and Volume data over a specified period.

3.2 Data Preprocessing

Normalization: Normalize features to ensure that all variables have similar scales, typically using Min-Max scaling (values between 0 and 1) for faster convergence.

Train-Test Split: Split data into training and testing datasets (e.g., 80-20 split).

Sliding Window Approach: Create input-output pairs using a sliding window over the time-series data. For example, to predict stock price at t+1t+1, use prices from t−nt−n to tt as input.

Reshaping Data for LSTM: LSTM expects a 3D input of shape (samples, time steps, features), so data is reshaped accordingly.

4. Model Architecture

4.1 LSTM Layers

The architecture consists of a sequential stack of LSTM layers. The number of layers and units can vary depending on the complexity of the data and model performance requirements.

Typical configurations:

Layer 1: LSTM layer with 50 units, return sequences=True (to stack another LSTM layer).

Layer 2: LSTM layer with 50 units.

Dropout Layers: To reduce overfitting, Dropout layers are added after each LSTM layer (e.g., dropout rate of 0.2).

4.2 Dense Layer

A fully connected layer is added at the end to map the output of LSTM layers to the stock price prediction.

4.3 Output Layer

A single-unit Dense layer to predict the stock price for the next time step.

5. Model Training

5.1 Loss Function

Mean Squared Error (MSE) is typically used as the loss function, as it penalizes large errors, which is beneficial for regression tasks like stock price prediction.

5.2 Optimization Algorithm

The model is optimized using the Adam optimizer, which is effective for handling sparse gradients and adapting learning rates.

5.3 Training Parameters

Batch Size: 64 (can be tuned)

Epochs: The number of epochs can range from 50 to 200, depending on the training time and convergence behaviour observed during experimentation.

6. Model Evaluation

6.1 Evaluation Metrics

Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are calculated on the test set to assess model accuracy.

6.2 Model Performance

Performance of the LSTM model is compared to a baseline model (e.g., linear regression or a simple moving average).

Visualization of predicted vs. actual stock prices helps analyse how well the model captures stock price trends and seasonality.

7. Results and Findings

7.1 Predictive Accuracy

If the LSTM model captures trends effectively, its predictions should closely align with the actual stock prices on the test set. However, sudden spikes or drops may be more challenging to predict.

7.2 Observations

LSTM models generally perform well in short-term predictions, but they may struggle with long-term predictions due to cumulative errors.

The model may need fine-tuning, including adjusting the number of layers, units, and training parameters, based on the specific stock's volatility and dataset size.

8. Conclusion

The LSTM-based model demonstrates the potential for predicting stock prices by capturing historical trends and patterns. However, due to inherent volatility and unpredictability in financial markets, the model may not always accurately predict sharp fluctuations. Fine-tuning and advanced architectures (e.g., attention-based LSTM models or hybrid models with convolutional layers) could further enhance predictive capabilities. Despite these challenges, LSTMs remain a powerful tool for time-series forecasting in financial applications.