**Assignment No: 4**

**Time Series Prediction Using Recurrent Neural Networks (RNNs)**

**Name: Shubham Gautam Dahane**

**PRN: 22310312**

**Roll: 382015**

**Problem Statement**

Implement time series prediction using **Recurrent Neural Networks (RNNs)** for stock market analysis. The goal is to predict Apple Inc. (AAPL) stock closing prices using historical data.

**Objectives**

* To understand the architecture and functioning of Recurrent Neural Networks.
* To preprocess time series data for RNN training.
* To implement an RNN model for stock market forecasting.
* To evaluate model performance on test data.
* To visualize predictions and compare them with actual values.

**Software & Hardware Used**

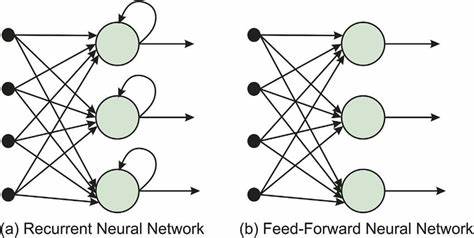
* **Operating System**: Windows/Linux/MacOS
* **Kernel**: Python 3.x
* **Tools**: Jupyter Notebook / Anaconda / Google Colab
* **Hardware**: CPU with 4 GB RAM (GPU optional for faster processing)

**Libraries Used**

* **TensorFlow / Keras** – model development
* **NumPy** – numerical operations
* **Pandas** – data handling
* **Matplotlib** – visualization
* **scikit-learn** – preprocessing (MinMaxScaler)
* **yfinance** – stock data acquisition

**Theory**

Recurrent Neural Networks (RNNs) are a class of neural networks designed for processing **sequential data**. They maintain a hidden state that captures temporal dependencies, making them suitable for time series forecasting.

* **Input Layer**: Accepts stock price sequences.
* **Recurrent Layers**: RNN cells (SimpleRNN, LSTM, GRU) capture short- and long-term dependencies.
* **Dense Layer**: Produces the final numeric prediction.
* **Activation Functions**: Tanh and Sigmoid are commonly used in recurrent cells.
* **Optimizers & Loss**: Adam optimizer with Mean Squared Error (MSE) loss is widely used.



**Methodology**

1. **Data Acquisition**
   * Collected Apple stock data (2015–2023) using Yahoo Finance.
   * Focused on the **closing price** as the target variable.
2. **Data Preprocessing**
   * Scaled stock prices into the range [0,1] using MinMaxScaler.
   * Created sequences of 60 days to predict the next day’s price.
3. **Train-Test Split**
   * Divided dataset into **80% training** and **20% testing**.
4. **Model Architecture**
   * Built an RNN model with two recurrent layers (SimpleRNN with 50 units each).
   * Added a Dropout layer for regularization.
   * Output layer used for regression prediction.
5. **Training**
   * Model compiled with Adam optimizer and MSE loss.
   * Trained for 20 epochs with batch size of 64.
6. **Evaluation**
   * Predictions generated on test dataset.
   * Results compared with actual stock prices.

Model Evaluation Metrics:

Mean Squared Error (MSE): 222.6845

Root Mean Squared Error (RMSE): 14.9226

Mean Absolute Error (MAE): 11.8189

R² Score: 0.6934

1. **Visualization**
   * Plotted predicted vs. actual stock prices to assess model performance.

A graph of blue and orange lines

AI-generated content may be incorrect.

**Advantages**

* RNNs effectively capture sequential dependencies.
* Memory units (LSTM/GRU) improve long-term learning.
* Flexible and adaptable to various time series domains.

**Limitations**

* Prone to **vanishing gradients** in long sequences (mitigated with LSTM/GRU).
* **Computationally intensive** training on large datasets.
* Risk of **overfitting** with small datasets.

**Applications**

* **Stock Market Analysis** – predicting future stock trends.
* **Weather Forecasting** – predicting temperature, rainfall, or humidity.
* **Natural Language Processing** – text generation, language modeling.

**Results**

* The RNN model successfully learned stock price patterns.
* Predictions followed the trend of actual stock prices with reasonable accuracy.
* Visualization confirmed that RNN can capture short-term fluctuations effectively.

**Conclusion**

Recurrent Neural Networks (RNNs) are a powerful tool for **time series forecasting**. By leveraging past stock price sequences, the model predicted future values with good accuracy. While RNNs capture short-term dependencies well, advanced architectures like **LSTM** and **GRU** are preferred for more complex, long-term forecasting tasks.

This assignment demonstrates that RNNs can be effectively applied to **financial data analysis** and **forecasting applications** in various domains.