# Assignment No:4

Practical Exercise: Time Series Prediction using RNN – Stock Market Analysis

## 1) Problem Statement

Implement time series prediction using Recurrent Neural Networks (RNN) for stock market analysis. The model should use historical stock prices to predict future prices, such as the next 20 days' stock prices, using the RNN architecture.

## 2) Libraries Used

Python:  
1. **TensorFlow/Keras**: For building and training the Recurrent Neural Network (RNN) model.  
2. **Pandas**: For data manipulation and loading the stock price dataset.  
3. **NumPy**: For handling arrays and data preprocessing.  
4. **Matplotlib**: For visualizing the results.  
5. **Scikit-learn**: For scaling and normalizing the data.

## 3) Theory

Recurrent Neural Networks (RNNs) are designed to work with sequential data, making them ideal for time series prediction tasks like stock market analysis. Unlike feedforward neural networks, RNNs have connections that form directed cycles, allowing them to maintain a 'memory' of previous inputs.  
  
In stock market analysis, RNNs are used to model the dependencies in stock prices over time. By learning from historical stock prices, the RNN model can predict future prices based on these temporal dependencies.

## 4) Methods

1. **Data Preprocessing**:  
 - The stock price dataset (e.g., Tesla stock prices) is loaded, and the 'Close' price column is extracted.  
 - The data is normalized using MinMaxScaler to scale the values between 0 and 1 for better model performance.  
 - Sequences of past stock prices are created, with each sequence used to predict the stock price for the next day.  
   
2. **Model Architecture**:  
 - A SimpleRNN model is built using Keras, with 50 units and ReLU activation.  
 - The model consists of one RNN layer and a fully connected output layer with a single neuron to predict the stock price.  
   
3. **Training**:  
 - The model is trained using the Adam optimizer and mean squared error (MSE) loss function.  
 - The dataset is split into training and test sets, and the model is trained on the training set.  
   
4. **Prediction and Evaluation**:  
 - The trained model is used to predict stock prices for the next 20 days, based on the last 60 days of historical data.  
 - The predicted prices are compared to the actual stock prices to evaluate the model's accuracy.

## 5) Advantages and Disadvantages

- **Advantages**:  
 - RNNs are well-suited for time series data and can capture temporal dependencies.  
 - TensorFlow/Keras makes it easy to build and train RNN models for time series prediction.  
   
- **Disadvantages**:  
 - RNNs may struggle with long-term dependencies and require more advanced architectures (e.g., LSTM, GRU) for better performance on complex datasets.  
 - Stock prices are influenced by various external factors, which may not be captured in a purely time series-based model.

## 6) Diagram

A computer screen shot of a computer

Description automatically generated

## 7) Conclusion

Recurrent Neural Networks (RNNs) provide a powerful tool for time series prediction tasks such as stock market analysis. By leveraging the temporal dependencies in historical stock prices, RNN models can make informed predictions about future stock prices. However, to achieve higher accuracy, more advanced RNN architectures like LSTMs may be needed, especially for complex and noisy datasets.