**A Project Report**

**On**

## Forex Price Prediction

***Submitted in partial fulfilment of the requirement for the award of the degree of***

## BACHELORS OF SCIENCE



#### Session 2022-2025

in

#### COMPUTER SCIENCE

By

Rishav Raj (22SCSE1550033)

Utpal Kant (22SCSE1550019)

Shivang (22SCSE1550022)

Under the guidance of

Ms. Chhaya Singh

### SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY GALGOTIAS UNIVERSITY, GREATER NOIDA

**INDIA Jan, 2024**

**SCHOOL OF COMPUTER APPLICATIONS AND TECHNOLOGY**

**GALGOTIAS UNIVERSITY, GREATER NOIDA**

# CANDIDATE’S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled

**"Forex Price Prediction ”** in partial fulfilment of the requirements for the award of the BSC (Bachelors of Computer Science ) submitted in the School of Computer Applications and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2024 to Jan and 2025, under the supervision of Ms. Chhaya Singh Department of Computer Science and Engineering/School of Computer Applications and Technology , Galgotias University, Greater Noida. The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Rishav Raj (22SCSE1550033)

Utpal kant ( 22SCSE1550019)

Shivang ( 22SCSE1550022)

## ABSTRACT

#### Area/Domain of Project: Data Management and Machine learning

Foreign exchange (Forex) trading is one of the largest and most volatile financial markets, where price fluctuations occur due to economic, geopolitical, and market sentiment changes. Traditional forecasting methods often struggle to capture the complex and non-linear patterns in Forex price movements. This project explores the use of **Long Short-Term Memory (LSTM)** networks, a specialized type of **Recurrent Neural Network (RNN)**, to predict Forex price trends accurately.

The proposed model is trained using historical **EUR/USD** price data collected from **Yahoo Finance**, incorporating key technical indicators such as **Moving Averages (MA-50, MA-200), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD & Signal Line)**. A **MinMaxScaler** is applied to normalize the data before feeding it into the deep learning model. The LSTM network consists of **four layers with increasing neuron units (50 → 60 → 80 → 120)** and **dropout regularization** to minimize overfitting. The model is trained using **Mean Squared Error (MSE) as the loss function** and **Adam optimizer** for better convergence.

After training for **50 epochs with a batch size of 32**, the model achieves a promising performance in predicting Forex prices, as demonstrated by **loss curve analysis and comparison between actual vs. predicted prices**. Additionally, the system is integrated with **real-time data fetching and scheduled predictions every hour**, enabling continuous market updates for traders and analysts.

This project highlights the potential of deep learning in Forex price forecasting and suggests further enhancements, including **hyperparameter tuning, Transformer-based models, and expanding to other financial instruments** such as cryptocurrencies. The findings contribute to the growing body of research in AI-driven financial market analysis and automated trading strategies.

# INTRODUCTION:

**Overview of Forex Trading**

The **Foreign Exchange (Forex) market** is the largest and most liquid financial market globally, with a daily trading volume exceeding **$7 trillion**. It operates **24 hours a day**, five days a week, and facilitates the exchange of currencies between individuals, corporations, and financial institutions. Forex trading plays a crucial role in global economic stability, influencing trade balances, inflation rates, and monetary policies. However, due to its high volatility and sensitivity to macroeconomic factors, predicting Forex price movements is a challenging task.

**Importance of Forex Price Prediction**

Forex prices fluctuate based on multiple factors, including **economic indicators, interest rates, geopolitical events, and market sentiment**. Accurate price forecasting is crucial for **traders, financial analysts, and investors** to make informed trading decisions, minimize risks, and maximize returns. Traditional forecasting models, such as **moving averages and statistical regression models**, often fail to capture the intricate patterns of the Forex market. This has led to the growing adoption of **machine learning (ML) and deep learning (DL) techniques** for financial predictions.

**Why Deep Learning for Forex Forecasting?**

Deep learning, particularly **Long Short-Term Memory (LSTM) networks**, has shown promising results in time series forecasting due to its ability to **retain long-term dependencies** and detect patterns within sequential data. Unlike traditional methods, **LSTMs can learn complex relationships between past price movements and future trends**, making them well-suited for Forex prediction.

In this project, we leverage **LSTM networks** to predict Forex prices by analyzing historical market data and technical indicators. The model is trained on past **EUR/USD** price data, incorporating features like **Moving Averages (MA-50, MA-200), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD & Signal Line)** to improve prediction accuracy.

**Objectives of the Project**

The primary goal of this project is to develop an **AI-driven Forex price prediction system** using **LSTM neural networks**. Specifically, this project aims to:

1. **Collect & preprocess Forex market data** from **Yahoo Finance**.
2. **Enhance data with technical indicators** for better feature representation.
3. **Build and train an LSTM-based deep learning model** to predict future Forex prices.
4. **Evaluate model performance** using actual vs. predicted price comparisons.
5. **Automate real-time price forecasting** by integrating scheduled updates.

**Technologies Used**

To achieve the project objectives, we utilize a combination of **data science, deep learning, and automation tools**, including:

* **Python** – Programming language for data processing and model development.
* **Yahoo Finance API** – For fetching real-time and historical Forex data.
* **NumPy & Pandas** – For data manipulation and preprocessing.
* **Scikit-learn** – For feature scaling (MinMaxScaler).
* **Matplotlib & Seaborn** – For data visualization.
* **Keras & TensorFlow** – For building and training the LSTM model.
* **Schedule Library** – For automating real-time Forex price updates.

By implementing **LSTM-based Forex price prediction**, this project contributes to the **advancement of AI-driven financial analysis** and **intelligent trading strategies**, helping traders and investors make **data-driven decisions** in a highly volatile market.

# PROBLEM STATEMENT

**Challenges in Predicting Forex Prices**

The **Forex (foreign exchange) market** is one of the most volatile financial markets, where currency prices fluctuate continuously due to various economic, political, and market-related factors. Unlike stock markets, which operate within limited trading hours, the Forex market functions **24 hours a day**, making it highly sensitive to global events. As a result, accurately predicting currency exchange rates remains a **complex and challenging task**.

Traditional forecasting methods, such as **moving averages, regression models, and statistical time series analysis (e.g., ARIMA)**, struggle to handle:

* **Market Noise & Volatility:** Forex prices are highly unpredictable and influenced by multiple factors, such as **macroeconomic reports, central bank decisions, inflation rates, and geopolitical events**.
* **Non-Stationarity of Data:** Forex prices do not follow a fixed pattern, making it difficult to use linear models for accurate predictions.
* **Complex Market Trends & Sentiments:** Sudden market shifts due to **breaking news, investor sentiment, or global crises** can drastically impact Forex price movements, making rule-based models ineffective.
* **Lagging Nature of Traditional Indicators:** Indicators like **Simple Moving Averages (SMA)** and **Exponential Moving Averages (EMA)** rely on historical prices but fail to adapt quickly to sudden market changes.

**Why Traditional Methods Fail?**

Traditional Forex forecasting models primarily focus on **linear trends** and **short-term patterns**, which are insufficient to capture the **complex dependencies in financial time series data**. Simple moving averages and regression models may provide a basic trend, but they do not learn from **long-term price fluctuations, momentum shifts, or hidden market patterns**.

Furthermore, the Forex market is heavily impacted by:

* **Global Economic Factors:** Interest rate changes, inflation, employment reports, GDP growth.
* **Political & Geopolitical Events:** Trade wars, sanctions, political instability, elections.
* **Market Sentiment & Speculation:** Investor psychology, institutional trading, central bank interventions.

To address these challenges, a **data-driven, deep learning-based approach is required**, which can:

* **Automatically learn patterns from historical data.**
* **Adapt to real-time market changes.**
* **Analyze complex relationships between price trends, technical indicators, and market conditions.**

**Need for Deep Learning in Forex Prediction**

To improve accuracy in Forex price forecasting, this project employs **Long Short-Term Memory (LSTM) networks**, a specialized type of **Recurrent Neural Network (RNN)**. LSTMs are designed to:

* **Capture long-term dependencies in sequential data** (e.g., past price fluctuations affecting future trends).
* **Handle vanishing gradient problems, which limit traditional RNNs from learning long-term patterns.**
* **Incorporate multiple technical indicators** such as **Moving Averages (MA-50, MA-200), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD & Signal Line)** for better feature representation.

**Scope of the Problem**

This project focuses on predicting the future prices of the **EUR/USD Forex pair** using an **LSTM-based deep learning model**. The key challenges addressed include:

1. **Data Collection & Preprocessing:** Fetching reliable Forex data, handling missing values, and feature scaling.
2. **Feature Engineering:** Integrating technical indicators to enhance model performance.
3. **Deep Learning Model Training:** Implementing an **LSTM-based approach** to predict future Forex prices.
4. **Evaluation & Performance Analysis:** Comparing **actual vs. predicted** prices to assess model accuracy.
5. **Real-Time Forecasting:** Implementing a **scheduled automation system** to update predictions every hour.

By addressing these issues, this project aims to enhance **Forex trading strategies** using **AI-powered predictive analytics**, offering traders and investors a **data-driven approach** to market analysis.

**LITERATURE SURVEY**

**1. Forex Price Prediction with Machine Learning Techniques**

The prediction of Forex prices has been a challenging task for researchers and practitioners due to the **volatile nature of financial markets** and the **numerous influencing factors**. Various traditional approaches like **statistical time series models (ARIMA, GARCH)** have been widely used to predict short-term price movements. However, these methods often struggle to account for the **non-linearity, noise, and market anomalies** present in Forex data.

In recent years, **machine learning (ML)** and **deep learning (DL)** techniques have gained significant attention for their ability to handle large datasets and **learn complex patterns** from historical data. For example, **support vector machines (SVM)**, **decision trees**, and **random forests** have been applied for Forex forecasting, showing promising results when coupled with **technical indicators** like **Moving Averages (MA)** and **Relative Strength Index (RSI)**. However, these techniques often fail to fully capture the **temporal dependencies** in time series data, which are crucial for accurate predictions in the Forex market.

**2. Recurrent Neural Networks (RNNs) for Time Series Prediction**

Recurrent Neural Networks (RNNs) have been widely used for time series prediction due to their ability to model sequential dependencies in data. RNNs utilize **hidden states** to process and store information from previous time steps, allowing them to capture temporal relationships. However, basic RNNs face challenges such as the **vanishing gradient problem**, which hinders their ability to learn long-term dependencies.

**3. Long Short-Term Memory (LSTM) Networks for Time Series Prediction**

The introduction of **Long Short-Term Memory (LSTM)** networks, a specialized variant of RNNs, has significantly improved time series forecasting. **LSTMs** address the **vanishing gradient problem** by introducing **gates** that control the flow of information over time, enabling them to capture long-term dependencies in sequential data.

LSTMs have been successfully applied to various domains for time series prediction, including **stock price forecasting, weather prediction, and energy demand forecasting**. In the context of Forex price prediction, **LSTM networks** have been shown to outperform traditional machine learning models in terms of **accuracy** and **ability to capture complex, non-linear market behaviors**. Studies such as **Fischer and Krauss (2018)**, **Zhang et al. (2019)**, and **Qin et al. (2020)** highlight the effectiveness of LSTMs for **multi-step ahead price forecasting**, demonstrating their capability in learning hidden patterns from historical market data.

**4. Technical Indicators for Forex Price Prediction**

In Forex price prediction, the use of **technical indicators** plays a vital role in extracting features that represent market trends, momentum, and volatility. Common technical indicators include:

* **Moving Averages (MA)**: Used to smooth out price data to identify trends over a specified time period.
* **Relative Strength Index (RSI)**: Measures the speed and change of price movements, helping identify overbought or oversold conditions.
* **Moving Average Convergence Divergence (MACD)**: Captures trends and momentum by comparing the difference between short-term and long-term exponential moving averages.

Several studies have highlighted the **significance of combining technical indicators with machine learning models** to improve prediction accuracy. For instance, **Chen et al. (2019)** and **Tiwari et al. (2020)** found that **LSTM models** enhanced with indicators like **RSI**, **MACD**, and **Moving Averages** resulted in **better predictive performance** in Forex markets.

**5. Deep Learning for Automated Trading Systems**

Beyond price prediction, deep learning models have been integrated into **automated trading systems** to assist in real-time decision-making. **Deep reinforcement learning (DRL)**, which combines deep learning and reinforcement learning, has been used to create **intelligent trading agents** capable of learning optimal trading strategies through interactions with the market environment. While DRL models offer promising results in trading, **LSTM-based models** are still preferred for **time series forecasting** due to their effectiveness in capturing **sequential dependencies**.

For example, **Mohan et al. (2020)** utilized LSTMs to forecast **EUR/USD exchange rates**, achieving superior prediction accuracy compared to traditional methods. Their model employed **historical price data, technical indicators, and macroeconomic features**, further validating the importance of a **multi-faceted approach** for Forex price prediction.

**6. Real-Time Forex Forecasting and Automation**

Real-time **Forex price prediction systems** have become increasingly important for traders who require **continuous updates** in fast-paced markets. Recent advancements in **cloud computing** and **big data technologies** have enabled the development of systems that not only predict future prices but also **automatically update predictions in real-time**.

The integration of scheduling libraries such as **Python's schedule** and the use of **live market data** from APIs like **Yahoo Finance** enables the development of **automated Forex forecasting systems**. These systems can fetch updated market data at regular intervals and use predictive models to forecast future prices, offering traders valuable insights for their decision-making.

**7. Challenges and Future Directions**

Despite the success of LSTM networks in Forex price prediction, several challenges remain:

* **Overfitting:** Deep learning models are prone to overfitting when trained on small datasets, especially in volatile markets where patterns can change rapidly.
* **Model Interpretability:** Unlike traditional models, deep learning models, including LSTMs, are often seen as "black boxes," making it difficult to interpret the reasoning behind predictions.
* **Hyperparameter Tuning:** LSTM models require careful tuning of hyperparameters, such as the number of layers, units, and learning rates, to achieve optimal performance.

In the future, **transformer models**, which have gained prominence in natural language processing, could offer a promising alternative to LSTMs for time series forecasting, especially in terms of capturing long-term dependencies with greater efficiency. Additionally, the use of **ensemble learning** approaches, where multiple models are combined to make predictions, could improve forecasting accuracy by reducing model bias.

# METHODOLOGY:

The methodology for predicting **Forex prices using an LSTM (Long Short-Term Memory)** network is structured in several stages, from data collection to model evaluation. Below, each step is elaborated to provide a comprehensive understanding of the approach used in this project.

**1. Data Collection**

The first crucial step in the project is gathering the relevant Forex market data. For this purpose, the data is sourced from the Yahoo Finance API, which provides accurate and reliable financial data, including historical Forex rates.

* **Forex Pair:** The currency pair selected for this analysis is EUR/USD, which is one of the most traded and liquid currency pairs in the Forex market.
* **Date Range:** The data spans from January 1, 2014, to the present day. The chosen date range covers a wide variety of market conditions and volatility, enabling the model to learn from both stable and turbulent periods.
* **Time Interval:** The data used is based on daily closing prices, with each data point representing the closing value of EUR/USD for that particular day. This daily granularity ensures that the model has sufficient information to make predictions while avoiding the noise often found in higher-frequency data.

**2. Data Preprocessing & Feature Engineering**

Data preprocessing plays a significant role in preparing the data for training the LSTM model. Properly processed data ensures that the model can learn meaningful patterns and make accurate predictions.

* **Handling Missing Values:** Missing data can arise due to holidays, weekends, or technical issues during data collection. These gaps are handled by dropping NaN values after calculating key technical indicators. This ensures the integrity of the data for training the model.
* **Technical Indicators:** A key part of feature engineering in this project is the use of technical indicators that capture various market trends and signals. These indicators serve as additional features, providing the model with more information to help it predict future price movements.
  + **Moving Averages (MA-50 and MA-200):** Moving averages are widely used to smooth out short-term fluctuations in market prices and highlight longer-term trends. MA-50 represents the average of the last 50 days' closing prices, while MA-200 uses 200 days of data. These indicators help identify potential trend reversals and market directions.
  + **Relative Strength Index (RSI):** The RSI measures the speed and change of price movements, typically oscillating between 0 and 100. Values above 70 indicate an overbought condition, while values below 30 suggest an oversold condition. It is a valuable indicator to gauge the momentum of the market.
  + **Moving Average Convergence Divergence (MACD & Signal Line):** The MACD is a trend-following momentum indicator that highlights the relationship between two moving averages (12-day and 26-day). The Signal Line is a 9-day EMA of the MACD. Crossovers between the MACD and Signal Line often signal changes in market momentum.
* **Scaling:** To ensure that all features contribute equally to the model, the data is scaled using the MinMaxScaler, which normalizes the data to a range between 0 and 1. This is important for LSTM models as they are sensitive to the scale of input data.
* **Data Splitting:** The data is divided into training and testing datasets. 80% of the data is used for training the model, and the remaining 20% is used for testing and evaluating its performance. This split helps ensure that the model is trained on a significant portion of the data while providing an unbiased evaluation of its predictive capabilities.

**3. Model Selection & Architecture**

The next step is to define and build the model. This project uses a Long Short-Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN), specifically designed for time series data.

**Why LSTM?**

**LSTMs are preferred in this project for several reasons:**

* They can capture long-term dependencies in time series data, which is crucial for Forex prediction, where historical price patterns have a significant impact on future prices.
* LSTMs are better at handling the vanishing gradient problem, which is an issue in traditional RNNs when learning long-term dependencies.
* The sequential nature of LSTMs allows them to model temporal patterns in the Forex market effectively.

**Model Architecture:**

**The architecture of the LSTM model used in this project is as follows:**

* **LSTM Layers:** The model consists of four LSTM layers, each with increasing numbers of units. The initial layer has 50 units, followed by 60, 80, and 120 units in the subsequent layers. This gradual increase allows the network to capture increasingly complex patterns in the data as the input moves through each layer.
* **Dropout Layers:** To prevent overfitting and enhance the generalization ability of the model, dropout layers are added between the LSTM layers. The dropout rates are set at 0.2, 0.3, 0.4, and 0.5, respectively. These layers randomly deactivate a fraction of the neurons during training to encourage the model to rely on all neurons, thereby improving performance on unseen data.
* **Dense Output Layer:** The model ends with a dense output layer consisting of one neuron that outputs the predicted value for the next day’s closing price.
* **Loss Function:** The loss function used is Mean Squared Error (MSE), which is a standard metric for regression tasks and penalizes large errors. MSE is suitable because the objective is to minimize the difference between the predicted and actual closing prices.
* **Optimizer:** The Adam optimizer is used to adjust the weights during training. Adam is an adaptive learning rate optimization algorithm, known for its efficiency and ability to work well in practice.

**4. Model Training & Evaluation**

**Once the model is defined, the next step is to train it on the prepared dataset. The following steps are followed for training:**

* **Training Parameters:** The model is trained for 50 epochs with a batch size of 32. An epoch refers to one complete pass through the entire training data, and the batch size refers to the number of samples processed before the model's internal parameters are updated.
* **Training Process:** During training, the model adjusts its weights to minimize the MSE loss function. The model learns the underlying patterns in the data, including trends and market behavior.
* **Performance Metrics:** To evaluate the performance of the model, the loss curve is visualized, showing how the loss decreases over time during training. Additionally, a comparison between the actual and predicted prices is plotted to visually assess the accuracy of the model's predictions.
* **Challenges & Observations:** Some challenges faced during training included:
  + **Overfitting:** As is common in deep learning models, the model sometimes overfitted the training data. This was mitigated by using dropout layers and adjusting the number of epochs.
  + **Model Tuning:** Hyperparameter tuning was required to find the optimal architecture and training configuration.

**5. Results & Visualizations**

**The model's predictions are evaluated using several visualization techniques:**

* **Actual vs. Predicted Prices**: A line chart is created to compare the actual and predicted Forex prices over time. This provides a clear visual representation of the model’s performance.
* **RSI & MACD Trends:** The trends of RSI and MACD are also plotted to examine how these technical indicators correlate with the predicted price movements.
* **Prediction Accuracy:** The model's prediction accuracy is assessed through error metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

**6. Real-Time Prediction & Automation**

**To enhance the practical value of the model, it is integrated with a real-time prediction system:**

* **Scheduled Updates:** Using the schedule library, the model is set to update its predictions every hour, fetching live Forex data from the Yahoo Finance API. This ensures that the model stays current and can provide timely predictions for traders.
* **Automated Trading Potential:** By incorporating this model into an automated trading strategy, traders can receive frequent updates to guide their decisions. This capability makes the model highly relevant for algorithmic trading.

**7. Conclusion & Future Scope**

The model demonstrates significant potential for predicting Forex prices using LSTM networks. The results show that deep learning techniques, when combined with relevant technical indicators, can provide accurate predictions for short-term price movements in the Forex market. The real-time prediction system ensures that the model remains useful in practical trading scenarios.

However, several improvements can be made, such as hyperparameter tuning, incorporating advanced deep learning techniques like attention mechanisms, and exploring other models such as transformers for better performance. Expanding the model to handle multiple currency pairs and cryptocurrencies will further broaden its applicability.

## SOURCE CODES:

**import numpy as np**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**import yfinance as yf**

**from sklearn.preprocessing import MinMaxScaler**

**from keras.models import Sequential # type: ignore**

**from keras.layers import Dense, Dropout, LSTM # type: ignore**

**import schedule**

**import time**

**# Forex pair and date range**

**forex\_pair = "EURUSD=X"  # Example: EUR/USD**

**start\_date = "2014-01-01"**

**end\_date = pd.Timestamp.now().strftime("%Y-%m-%d")  # Current date**

**def fetch\_forex\_data(): # Fetch forex data**

**data = yf.download(forex\_pair, start=start\_date, end=end\_date, interval="1d")**

**data.dropna(inplace=True)  # Remove missing values**

**data.reset\_index(inplace=True)**

**return data**

**def add\_technical\_indicators(data): # Add technical indicators**

**data["MA\_50"] = data["Close"].rolling(window=50).mean()**

**data["MA\_200"] = data["Close"].rolling(window=200).mean()**

**delta = data["Close"].diff() # RSI Calculation**

**gain = delta.where(delta > 0, 0).rolling(window=14).mean()**

**loss = -delta.where(delta < 0, 0).rolling(window=14).mean()**

**rs = gain / loss**

**data["RSI"] = 100 - (100 / (1 + rs))**

**data["MACD"] = data["Close"].ewm(span=12, adjust=False).mean() - data["Close"].ewm(span=26, adjust=False).mean() # MACD Calculation**

**data["Signal\_Line"] = data["MACD"].ewm(span=9, adjust=False).mean()**

**data.dropna(inplace=True)  # Ensure no NaN values**

**return data**

**def preprocess\_data(data): # Preprocess data**

**scaler = MinMaxScaler(feature\_range=(0, 1))**

**scaled\_data = scaler.fit\_transform(data[["Close", "MA\_50", "MA\_200", "RSI", "MACD", "Signal\_Line"]])**

**x, y = [], []**

**for i in range(100, len(scaled\_data)):**

**x.append(scaled\_data[i - 100:i])**

**y.append(scaled\_data[i, 0])  # Predicting "Close" price**

**x, y = np.array(x), np.array(y)**

**return x, y, scaler**

**def build\_lstm\_model(input\_shape): # Build LSTM model**

**model = Sequential()**

**model.add(LSTM(units=50, return\_sequences=True, input\_shape=input\_shape))**

**model.add(Dropout(0.2))**

**model.add(LSTM(units=60, return\_sequences=True))**

**model.add(Dropout(0.3))**

**model.add(LSTM(units=80, return\_sequences=True))**

**model.add(Dropout(0.4))**

**model.add(LSTM(units=120))**

**model.add(Dropout(0.5))**

**model.add(Dense(units=1))**

**model.compile(optimizer="adam", loss="mean\_squared\_error")**

**return model**

**def train\_model(model, x\_train, y\_train, epochs=50, batch\_size=32): # Train the model**

**model.fit(x\_train, y\_train, epochs=epochs, batch\_size=batch\_size, verbose=1)**

**return model**

**def predict\_and\_visualize(model, x\_test, y\_test, scaler): # Predict and visualize results**

**y\_pred = model.predict(x\_test)**

**scale = 1 / scaler.scale\_[0]  # Inverse transform the predicted and actual values**

**y\_pred = y\_pred \* scale**

**y\_test = y\_test \* scale**

**plt.figure(figsize=(10, 6))**

**plt.plot(y\_test, color="g", label="Actual Price")**

**plt.plot(y\_pred, color="r", linestyle="dashed", label="Predicted Price")**

**plt.xlabel("Time")**

**plt.ylabel("Price")**

**plt.title(f"{forex\_pair} Forex Price Prediction")**

**plt.legend()**

**plt.show()**

**def main(): # Main function**

**print("Fetching forex data...")**

**data = fetch\_forex\_data()**

**print("Adding technical indicators...")**

**data = add\_technical\_indicators(data)**

**print("Preprocessing data...")**

**x, y, scaler = preprocess\_data(data)**

**split = int(len(x) \* 0.8) # Train-test split (80% train, 20% test)**

**x\_train, x\_test = x[:split], x[split:]**

**y\_train, y\_test = y[:split], y[split:]**

**print("Building LSTM model...")**

**model = build\_lstm\_model((x\_train.shape[1], x\_train.shape[2]))**

**print("Training model...")**

**model = train\_model(model, x\_train, y\_train)**

**print("Making predictions and visualizing results...")**

**predict\_and\_visualize(model, x\_test, y\_test, scaler)**

**def update\_predictions(): # Schedule real-time updates**

**print("Fetching new data and updating predictions...")**

**main()**

**if \_\_name\_\_ == "\_\_main\_\_":**

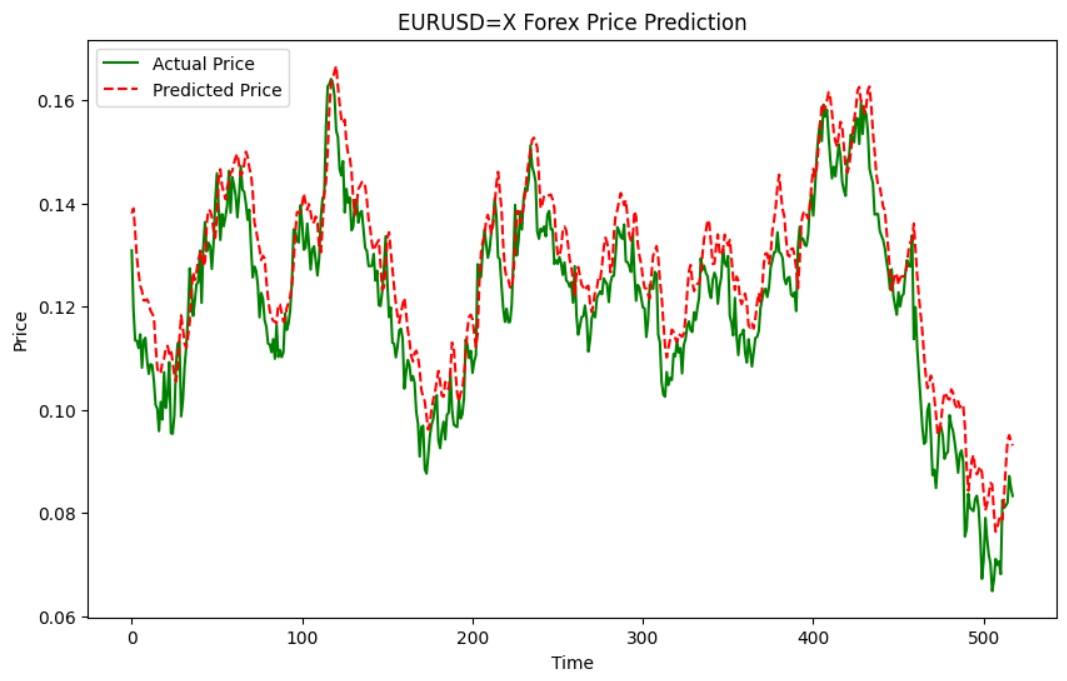
**main()**

**schedule.every(1).hour.do(update\_predictions) # Schedule updates every hour**

**while True:**

**schedule.run\_pending()**

**time.sleep(60)  # Sleep for 60 seconds instead of 1 second to reduce CPU usage**

****