VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**TRẦN CỰ PHÚ - 520H0667**

**TRẦN HẢI QUANG – 520H0673**

EXCHANGE RATE PREDICTION USING DEEP LEARNING MODELS

**INFORMATION TECHNOLOGY PROJECT 2**

**COMPUTER SCIENCE**

**HO CHI MINH, 2024**

VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**TRẦN CỰ PHÚ - 520H0667**

**TRẦN HẢI QUANG – 520H0673**

EXCHANGE RATE PREDICTION USING DEEP LEARNING MODELS

**INFORMATION TECHNOLOGY PROJECT 2**

**COMPUTER SCIENCE**

Advised by

**TS. Trịnh Hùng Cường**

**HO CHI MINH, 2024**

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to Mr. Trinh Hung Cuong, our instructor and mentor, for his valuable guidance and support throughout the report of our project on predicting exchange rate using deep learning models. He has been very helpful and patient in providing us with constructive feedback and suggestions to improve our work. He has also encouraged us to explore new technologies and techniques to enhance our model’s functionality and performance. We have learned a lot from his expertise and experience in web development and software engineering. We are honored and privileged to have him as our teacher and supervisor.

*TP. Ho Chi Minh, January 4th, 2024*

*Writer*

*(Sign and write full name)*

***Quang***

Tran Hai Quang

***Phu***

Tran Cu Phu

**DECLARATION OF AUTHORSHIP**

We hereby declare that this is our own project and is guided by Mr. Trinh Hung Cuong; The content research and results contained herein are central and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the main author from different sources, which are clearly stated in the reference section.

In addition, the project also uses some comments, assessments as well as data of other authors, other organizations with citations and annotated sources.

If something wrong happens, We’ll take full responsibility for the content of my project. Ton Duc Thang University is not related to the infringing rights, the copyrights that We give during the implementation process (if any).

*TP. Ho Chi Minh, January 4th, 2024*

*Writer*

*(Sign and write full name)*

***Quang***

Tran Hai Quang

***Phu***

Tran Cu Phu

**EXCHANGE RATE PREDICTING USING DEEP LEARNING MODELS**

**ABSTRACT**

This report offers a timely exploration of contemporary methods employed in predicting exchange rates amidst the rapidly evolving landscape of global financial markets. As financial technology continues to advance, researchers and practitioners are exploring innovative techniques to enhance the accuracy and reliability of exchange rate prediction.

The research process initiates with the collection of historical data regarding open price, close price, high price, low price and volume spanning a decade from 01/01/2014 to 01/01/2024, for the following pairs: EUR/USD, GBP/USD, USD/CAD. Subsequently, the data undergoes preprocessing to eliminate noise and missing values, while being transformed into a suitable format for model training. A variety of machine learning models such as Recurrent Neural Network, Long Short-Term Memory, Gated Recurrent Unit, Artificial Neural Network and Multi-layer Perceptron are constructed.

An emphasis is placed on real-time prediction capabilities, acknowledging the increasing importance of up-to-the-minute information in the foreign exchange market. The report discusses the challenges associated with modeling in a dynamic environment and explores strategies for adapting predictive models to capture sudden market shifts and unprecedented events.

The report extensively elaborates on the research and analytical steps, encompassing data preprocessing, model construction and performance evaluation. Nonetheless, it is imperative to acknowledge that predicting exchange rate is a complex undertaking and outcomes may be influenced by numerous unforeseeable factors

**TABLE OF CONTENT**

[LIST OF FIGURES vi](#_Toc160650540)

[ABBREVIATIONS x](#_Toc160650541)

[CHAPTER 1. OVERVIEW 1](#_Toc160650542)

[1.1 Foreign Exchange Market Introduction 1](#_Toc160650543)

[1.2 Introduction 2](#_Toc160650544)

[CHAPTER 2. THEORETICAL BASIS 3](#_Toc160650545)

[2.1 Deep Learning Models 3](#_Toc160650546)

[2.1.1 Recurrent Neural Network 3](#_Toc160650547)

[2.1.2 Long Short-Term Memory 4](#_Toc160650548)

[2.1.3 Gated Recurrent Unit 6](#_Toc160650549)

[2.1.4 Artificial Neural Network 7](#_Toc160650550)

[2.1.5 Multi-Layer-Perceptron 8](#_Toc160650551)

[CHAPTER 3. DATA COLLECTING 10](#_Toc160650552)

[3.1 Dataset 10](#_Toc160650553)

[3.2 Dataset Detail 10](#_Toc160650554)

[3.3 Collecting Data 11](#_Toc160650555)

[3.4 Data Splitting 12](#_Toc160650556)

[CHAPTER 4. PREDICTING PRICE USING DEEP LEARNING MODELS 14](#_Toc160650557)

[*4.1* Recurrent Neural Network Model 14](#_Toc160650558)

[4.1.1 EUR/USD Prediction Using RNN 14](#_Toc160650559)

[4.1.2 GBP/USD Prediction Using RNN 17](#_Toc160650560)

[4.1.3 USD/CAD Prediction Using RNN 20](#_Toc160650561)

[*4.2* Long Short-Term Memory Model 22](#_Toc160650562)

[4.2.1 EUR/USD Prediction Using LSTM 23](#_Toc160650563)

[4.2.2 GBP/USD Prediction Using LSTM 26](#_Toc160650564)

[4.2.3 USD/CAD Prediction Using LSTM 28](#_Toc160650565)

[*4.3* Gated Recurrent Unit Model 31](#_Toc160650566)

[4.3.1 EUR/USD Prediction Using GRU 32](#_Toc160650567)

[4.3.2 GBP/USD Prediction Using GRU 34](#_Toc160650568)

[4.3.3 USD/CAD Prediction Using GRU 37](#_Toc160650569)

[*4.4* Artificial Neural Network Model 39](#_Toc160650570)

[4.4.1 EUR/USD Prediction Using ANN 40](#_Toc160650571)

[4.4.2 GBP/USD Prediction Using ANN 43](#_Toc160650572)

[4.4.3 USD/CAD Prediction Using ANN 45](#_Toc160650573)

[*4.5* Multi-layer Perceptron Model 48](#_Toc160650574)

[4.5.1 EUR/USD Prediction Using MLP 48](#_Toc160650575)

[4.5.2 GBP/USD Prediction Using MLP 51](#_Toc160650576)

[4.5.3 USD/CAD Prediction Using MLP 54](#_Toc160650577)

[CHAPTER 5. EXCHANGE RATE PREDICTION WEBSITE DEMO 57](#_Toc160650578)

[5.1 Setup 57](#_Toc160650579)

[5.2 Demo 62](#_Toc160650580)

[CHAPTER 6: CONCLUSION 67](#_Toc160650581)

[6.1 Conclusion 67](#_Toc160650582)

[6.2 Development Direction 68](#_Toc160650583)

[REFERENCES 69](#_Toc160650584)

# LIST OF FIGURES

[Figure 3.1: Download data 11](#_Toc160648680)

[Figure 3.2: Convert data to X, y function 12](#_Toc160648681)

[Figure 3.3: Convert data to X, y 12](#_Toc160648682)

[Figure 3.4: Normalize the X dataset 13](#_Toc160648683)

[Figure 3.5: Split data into train and test 13](#_Toc160648684)

[Figure 4.1: RNN Model 14](#_Toc160648685)

[Figure 4.2: RNN Prediction on EUR/USD Train data 14](#_Toc160648686)

[Figure 4.3: Visualize the RNN prediction on EUR/USD Train data 15](#_Toc160648687)

[Figure 4.4: RNN Prediction on EUR/USD Test data 15](#_Toc160648688)

[Figure 4.5: Visualize the RNN prediction on EUR/USD Test data 16](#_Toc160648689)

[Figure 4.6: Visualize the RNN prediction on EUR/USD 16](#_Toc160648690)

[Figure 4.7: Train RNN model using GBP/USD data 17](#_Toc160648691)

[Figure 4.8: RNN Prediction on GBP/USD Train data 17](#_Toc160648692)

[Figure 4.8: Visualize the RNN prediction on GBP/USD Train data 18](#_Toc160648693)

[Figure 4.9: RNN Prediction on GBP/USD Test data 18](#_Toc160648694)

[Figure 4.10: Visualize the RNN prediction on GBP/USD Test data 19](#_Toc160648695)

[Figure 4.11: Visualize the RNN prediction on GBP/USD 19](#_Toc160648696)

[Figure 4.12: Train RNN model using USD/CAD data 20](#_Toc160648697)

[Figure 4.13: RNN Prediction on USD/CAD Train data 20](#_Toc160648698)

[Figure 4.14: Visualize the RNN prediction on USD/CAD Train data 20](#_Toc160648699)

[Figure 4.15: RNN Prediction on USD/CAD Test data 21](#_Toc160648700)

[Figure 4.16: Visualize the RNN prediction on USD/CAD Test data 21](#_Toc160648701)

[Figure 4.17: Visualize the RNN prediction on USD/CAD 22](#_Toc160648702)

[Figure 4.18: LSTM Model 23](#_Toc160648703)

[Figure 4.19: LSTM Prediction on EUR/USD Train data 23](#_Toc160648704)

[Figure 4.20: Visualize the LSTM prediction on EUR/USD Train data 24](#_Toc160648705)

[Figure 4.21: LSTM Prediction on Test data 24](#_Toc160648706)

[Figure 4.22: Visualize the LSTM prediction on EUR/USD Test data 25](#_Toc160648707)

[Figure 4.23: Visualize the LSTM prediction on EUR/USD 25](#_Toc160648708)

[Figure 4.24: Train LSTM model using GBP/USD data 26](#_Toc160648709)

[Figure 4.25: LSTM Prediction on GBP/USD Train data 26](#_Toc160648710)

[Figure 4.26: Visualize the LSTM prediction on GBP/USD Train data 26](#_Toc160648711)

[Figure 4.27: LSTM Prediction on GBP/USD Test data 27](#_Toc160648712)

[Figure 4.28: Visualize the LSTM prediction on GBP/USD Test data 27](#_Toc160648713)

[Figure 4.29: Visualize the LSTM prediction on GBP/USD 28](#_Toc160648714)

[Figure 4.30: Train LSTM model using USD/CAD data 28](#_Toc160648715)

[Figure 4.31: LSTM Prediction on USD/CAD Train data 28](#_Toc160648716)

[Figure 4.32: Visualize the LSTM prediction on USD/CAD Train data 29](#_Toc160648717)

[Figure 4.33: LSTM Prediction on USD/CAD Test data 29](#_Toc160648718)

[Figure 4.34: Visualize the LSTM prediction on USD/CAD Test data 30](#_Toc160648719)

[Figure 4.35: Visualize the LSTM prediction onUSD/CAD 30](#_Toc160648720)

[Figure 4.36: GRU Model 31](#_Toc160648721)

[Figure 4.37: GRU Prediction on EUR/USD Train data 32](#_Toc160648722)

[Figure 4.38: Visualize the GRU prediction on EUR/USD Train data 32](#_Toc160648723)

[Figure 4.39: GRU Prediction on Test data 33](#_Toc160648724)

[Figure 4.40: Visualize the GRU prediction on EUR/USD Test data 33](#_Toc160648725)

[Figure 4.41: Visualize the GRU prediction on EUR/USD 34](#_Toc160648726)

[Figure 4.42: Train GRU model using GBP/USD data 34](#_Toc160648727)

[Figure 4.43: GRU Prediction on GBP/USD Train data 34](#_Toc160648728)

[Figure 4.44: Visualize the GRU prediction on GBP/USD Train data 35](#_Toc160648729)

[Figure 4.45: GRU Prediction on GBP/USD Test data 35](#_Toc160648730)

[Figure 4.46: Visualize the GRU prediction on GBP/USD Test data 36](#_Toc160648731)

[Figure 4.47: Visualize the GRU prediction on GBP/USD 36](#_Toc160648732)

[Figure 4.48: Train GRU model using USD/CAD data 37](#_Toc160648733)

[Figure 4.49: GRU Prediction on USD/CAD Train data 37](#_Toc160648734)

[Figure 4.50: Visualize the GRU prediction on USD/CAD Train data 37](#_Toc160648735)

[Figure 4.51: GRU Prediction on USD/CAD Test data 38](#_Toc160648736)

[Figure 4.52: Visualize the GRU prediction on USD/CAD Test data 38](#_Toc160648737)

[Figure 4.53: Visualize the GRU prediction on USD/CAD 39](#_Toc160648738)

[Figure 4.54: ANN Model 40](#_Toc160648739)

[Figure 4.55: ANN Prediction on EUR/USD Train data 40](#_Toc160648740)

[Figure 4.56: Visualize the ANN prediction on EUR/USD Train data 41](#_Toc160648741)

[Figure 4.57: ANN Prediction on EUR/USD Test data 41](#_Toc160648742)

[Figure 4.58: Visualize the ANN prediction on EUR/USD Test data 42](#_Toc160648743)

[Figure 4.59: Visualize the ANN prediction on EUR/USD 42](#_Toc160648744)

[Figure 4.60: Train ANN model using GBP/USD data 43](#_Toc160648745)

[Figure 4.61: ANN Prediction on GBP/USD Train data 43](#_Toc160648746)

[Figure 4.45: ANN Prediction on GBP/USD Test data 44](#_Toc160648747)

[Figure 4.46: Visualize the ANN prediction on GBP/USD Test data 44](#_Toc160648748)

[Figure 4.47: Visualize the ANN prediction on GBP/USD 45](#_Toc160648749)

[Figure 4.48: Train ANN model using USD/CAD data 45](#_Toc160648750)

[Figure 4.49: ANN Prediction on USD/CAD Train data 46](#_Toc160648751)

[Figure 4.50: Visualize the ANN prediction on USD/CAD Train data 46](#_Toc160648752)

[Figure 4.51: ANN Prediction on USD/CAD Test data 46](#_Toc160648753)

[Figure 4.52: Visualize the ANN prediction on USD/CAD Test data 47](#_Toc160648754)

[Figure 4.53: Visualize the ANN prediction on USD/CAD 47](#_Toc160648755)

[Figure 4.54: MLP Model 48](#_Toc160648756)

[Figure 4.55: MLP Prediction on EUR/USD Train data 48](#_Toc160648757)

[Figure 4.56: Visualize the MLP prediction on EUR/USD Train data 49](#_Toc160648758)

[Figure 4.57: MLP Prediction on EUR/USD Test data 49](#_Toc160648759)

[Figure 4.58: Visualize the MLP prediction on EUR/USD Test data 50](#_Toc160648760)

[Figure 4.59: Visualize the MLP prediction on EUR/USD 50](#_Toc160648761)

[Figure 4.60: Train MLP model using GBP/USD data 51](#_Toc160648762)

[Figure 4.61: MLP Prediction on GBP/USD Train data 51](#_Toc160648763)

[Figure 4.62: Visualize the MLP prediction on GBP/USD Train data 52](#_Toc160648764)

[Figure 4.63: MLP Prediction on GBP/USD Test data 52](#_Toc160648765)

[Figure 4.64: Visualize the MLP prediction on GBP/USD Test data 53](#_Toc160648766)

[Figure 4.65: Visualize the MLP prediction on GBP/USD 53](#_Toc160648767)

[Figure 4.66: Train MLP model using USD/CAD data 54](#_Toc160648768)

[Figure 4.67: MLP Prediction on USD/CAD Train data 54](#_Toc160648769)

[Figure 4.68: Visualize the MLP prediction on USD/CAD Train data 54](#_Toc160648770)

[Figure 4.69: MLP Prediction on USD/CAD Test data 55](#_Toc160648771)

[Figure 4.70: Visualize the MLP prediction on USD/CAD Test data 55](#_Toc160648772)

[Figure 4.71: Visualize the GRU prediction on USD/CAD 56](#_Toc160648773)

[Figure 5.1: model.py script 58](#_Toc160648774)

[Figure 5.2: currency.py script 60](#_Toc160648775)

[Figure 5.4: Run Streamlit app 61](#_Toc160648776)

[Figure 5.5: Retrieve Password/Enpoint IP 62](#_Toc160648777)

[Figure 5.6: Run localtunnel 62](#_Toc160648778)

[Figure 5.8: Select models 63](#_Toc160648779)

[Figure 5.9: Input prices 64](#_Toc160648780)

[Figure 5.11: Display prediction and visualization 66](#_Toc160648781)

# ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| GRU | Gated Recurrent Unit | |
| ANN | Artificial Neural Network | |
| MLP | Multi-Layer Perceptron | |
| ReLU | Rectified Linear Unit | |
| API | Application Programming Interface | |
| IP | Internet Protocol | |
|  |  | |
|  |  |
|  |  |
|  |  |
|  |  |

# CHAPTER 1. OVERVIEW

## Foreign Exchange Market Introduction

The foreign exchange market is an over-the-counter global marketplace that determines the exchange rate for currencies around the world. Participants in these markets can buy, sell, exchange, and speculate on the relative exchange rates of various currency pairs.

Foreign exchange markets are made up of banks, forex dealers, commercial companies, central banks, investment management firms, hedge funds, retail forex dealers, and investors.

The currency exchange market plays a crucial role in the global economy by facilitating the exchange of currencies:

- The forex market enables countries to engage in international trade by providing a mechanism for converting one currency into another. Businesses can buy and sell goods and services across borders, and the forex market ensures efficient currency conversion.

- Exchange rates, which are determined in the forex market, play a crucial role in a country's economic health. They affect the cost of imported goods, the competitiveness of exports, and, consequently, the balance of trade.

- The currency exchange market facilitates international tourism by allowing travelers to exchange their home currency for the currency of the destination country. This supports the tourism industry and contributes to the economy through spending by foreign visitors.

## Introduction

The foreign exchange market's influence on global economics necessitates accurate exchange rate predictions. In recent years, deep learning models have emerged as powerful tools to address the challenges associated with predicting currency movements.

Deep learning models autonomously learn features from historical exchange rate data, eliminating the need for manual feature engineering and enhancing their ability to recognize complex patterns. Deep learning models seamlessly integrate diverse datasets, enhancing their capacity to account for economic indicators. Advanced optimization algorithms and adaptive learning rates equip deep learning models to handle non-stationary exchange rate data, ensuring robust predictions in dynamic market conditions.

Deep learning models offer a promising avenue for enhancing exchange rate prediction. Their ability to adapt to dynamic patterns, integrate diverse datasets, and handle non-stationary data positions them as valuable tools for businesses, investors, and policymakers navigating the complexities of the foreign exchange market.

# CHAPTER 2. THEORETICAL BASIS

## Deep Learning Models

### Recurrent Neural Network

Recurrent Neural Networks are a type of artificial neural network designed to process sequential data by maintaining a memory or state of previous inputs. Unlike traditional feedforward neural networks, where the flow of information moves in one direction, from input to output, RNNs have connections that form directed cycles, allowing them to capture temporal dependencies in sequential data.

How RNNs Work:

- RNNs take a sequence of input data as their input. This sequence could be a time series, a sentence, or any other ordered set of data points.

- At each time step in the sequence, the RNN processes the current input along with the hidden state from the previous time step. The computation at each time step involves a weighted sum of the current input and the hidden state from the previous time step. This weighted sum is passed through an activation function to produce the new hidden state.

- RNNs use the same set of weights and biases across all time steps. This parameter sharing enables the network to learn and generalize patterns in the entire sequence. The network is trained using backpropagation through time, updating the weights to minimize the difference between predicted and actual outcomes.

Advantages of RNN:

- Well-suited for tasks involving sequential data, such as time series prediction, natural language processing, and speech recognition.

- Can handle input sequences of varying lengths due to parameter sharing across time steps.

- Capable of retaining information about past inputs through the hidden state, making them effective for tasks requiring memory.

Disadvantages of RNN:

- RNNs are prone to issues like vanishing and exploding gradients, especially in long sequences, which can hinder training.

- RNNs have difficulties in capturing long-term dependencies in sequences, and their ability to retain information over long distances is limited.

- Training RNNs can be computationally intensive, and the sequential nature of their processing makes them less parallelizable compared to other architectures.

### Long Short-Term Memory

Long Short-Term Memory is a type of recurrent neural network architecture designed to address the challenges of capturing and learning long-term dependencies in sequential data. Traditional RNNs, due to their vanishing or exploding gradient problems, struggle with effectively retaining information over long sequences. LSTMs, introduced by Hochreiter and Schmidhuber in 1997, overcome these challenges through a more complex and structured design.

How LSTM Work:

- LSTMs introduce the concept of memory cells, which are special units designed to store information over long periods. LSTMs have a more complex structure that allows them to capture and maintain information over extended sequences.

- The architecture of LSTMs addresses the vanishing gradient problem associated with traditional RNNs, allowing them to capture long-term dependencies more effectively. The input, forget, and output gates enable LSTMs to decide when to remember, forget, or output information, making them well-suited for tasks with extended temporal dependencies.

- The input, forget, and output gates enable LSTMs to decide when to remember, forget, or output information, making them well-suited for tasks with extended temporal dependencies.

Advantages of LSTM:

- LSTMs excel at capturing long-term dependencies in sequential data, addressing the vanishing gradient problem.

- Forget gate mechanism allows nuanced control over what to remember and forget, enhancing memory management.

- Effective in various tasks like time series prediction, natural language processing, and speech recognition.

Disadvantages of LSTM:

- LSTMs can be resource-intensive during training and may pose scalability challenges.

- Complexity hinders interpretability, making it challenging to understand the model's inner workings.

- Prone to overfitting, especially with limited or noisy training data.

### Gated Recurrent Unit

Gated Recurrent Unit is another type of recurrent neural network architecture, similar to LSTM, designed to address the challenges of learning long-term dependencies in sequential data. GRU simplifies the LSTM architecture by combining the memory cell and hidden state, using gating mechanisms to control the flow of information.

How GRU Work:

- GRUs introduce two gates, the update gate and the reset gate, which control the flow of information in and out of the hidden state. The update gate determines how much of the previous hidden state should be combined with the candidate new hidden state, while the reset gate determines how much of the past information to forget.

- GRUs use a candidate hidden state that is computed based on the current input and the previous hidden state modified by the reset gate. This candidate hidden state represents the information that might be added to the updated hidden state. The update gate then determines how much of the candidate state should be included in the new hidden state.

- The use of update and reset gates provides a mechanism for selectively retaining and updating information, allowing GRUs to capture long-term dependencies more effectively than standard RNNs.

Advantages of GRU:

- GRUs generally have fewer parameters compared to LSTMs, making them computationally more efficient and faster to train.

- GRUs are often easier to train compared to LSTMs, especially on smaller datasets, due to their simpler architecture.

- GRUs can effectively capture dependencies in sequential data, balancing the trade-off between modeling complex patterns and computational efficiency.

Disadvantages of GRU:

- GRUs may struggle with capturing very long-term dependencies compared to LSTMs, as they do not have a separate memory cell.

- LSTMs, with their separate memory cell, can have a higher capacity to capture nuanced long-term patterns compared to GRUs.

### Artificial Neural Network

Artificial Neural Network is a computational model inspired by the structure and functioning of the human brain. It consists of interconnected nodes, organized in layers, and is used for machine learning and artificial intelligence tasks. ANNs are designed to simulate the way biological neural networks work, allowing them to learn and make decisions based on input data.

How ANN Work:

- ANNs consist of interconnected nodes, also known as neurons or artificial neurons, organized into layers. The basic building block is a neuron, which takes multiple input values, applies weights to these inputs, sums them up, adds a bias, and then passes the result through an activation function. Layers can be categorized into an input layer, one or more hidden layers, and an output layer.

- ANNs operate in a feedforward manner, meaning that information flows through the network from the input layer to the hidden layers and, finally, to the output layer. Each layer's neurons contribute to the activation of neurons in the next layer, allowing the network to learn complex patterns and representations from the input data.

Advantages of ANN:

- ANNs can be applied to a wide range of tasks, including classification, regression, pattern recognition, and more.

- ANNs can adapt and learn from data, making them suitable for tasks with complex and non-linear relationships.

- Neural networks can process information in parallel, leading to faster training times and real-time predictions in certain scenarios.

Disadvantages of ANN:

- ANNs are often considered as black box models, making it challenging to interpret and understand their internal workings.

- Training large neural networks can be computationally intensive, requiring substantial resources, especially for deep architectures.

- ANNs typically require large amounts of labeled data for effective training, which might be a limitation in tasks with limited data availability.

### Multi-Layer-Perceptron

A multilayer perceptron is a name for a modern feedforward artificial neural network, consisting of fully connected neurons with a nonlinear kind of activation function, organized in at least three layers, notable for being able to distinguish data that is not linearly separable.

How MLP Work:

- An MLP consists of multiple layers of nodes (neurons), organized into three main types of layers:

* + - * + Input Layer: Neurons in this layer represent the features of the input data.
        + Hidden Layers: Intermediate layers that perform computations on the input data. Each neuron in a hidden layer applies an activation function to a weighted sum of its inputs.
        + Output Layer: Neurons in this layer produce the final output based on the computations in the hidden layers.

- Neurons in one layer are connected to neurons in the next layer through weighted connections. Each connection has an associated weight, which determines the strength of the connection. During training, these weights are adjusted using optimization algorithms to minimize the difference between the predicted output and the actual target output.

- Each neuron in the hidden and output layers typically applies an activation function to the weighted sum of its inputs. Common activation functions include the sigmoid, hyperbolic tangent, and rectified linear unit. Activation functions introduce non-linearity to the model, enabling it to learn complex patterns and relationships in the data.

Advantages of MLP:

- MLPs are versatile and can be applied to a wide range of tasks.

- The presence of non-linear activation functions allows MLPs to learn complex relationships in data.

- MLPs can automatically learn relevant features from the input data, reducing the need for manual feature engineering.

Disadvantages of MLP:

- Training large MLPs can be computationally intensive, requiring substantial resources.

- MLPs, especially deep architectures, may be prone to overfitting, particularly when dealing with small datasets.

- The complex nature of MLPs can make them challenging to interpret and understand, often being considered as black box models.

# CHAPTER 3. DATA COLLECTING

## Dataset

The dataset in this task has been collected from the currency pairs: EUR/USD, USD/CAD, GBP/USD spanning over a period of 10 years from 01/12/2003 to 21/12/2013.

The dataset comprises detailed information on the trading prices of the mentioned currency pairs during trading sessions. It includes open prices, close prices, high prices, low prices and volume.

Possessing an extensive and enduring dataset in the realm of foreign exchange market facilitates the analysis of prolonged trends, identification of intricate patterns, and comprehension of the repercussions of significant events on stock prices. Leveraging advanced deep learning techniques and predictive models empowers us to delve into the nuanced relationships among financial and technicalthereby enabling the anticipation of future price fluctuations.

## Dataset Detail

High Price: The highest traded price of a financial instrument during a specified time period, often within a trading day. It represents the peak value reached by the instrument during that period.

Low Price: The lowest traded price of a financial instrument during a specified time period, typically within a trading day. It represents the minimum value reached by the instrument during that period.

Open Price: The first traded price of a financial instrument at the beginning of a specific time period, such as the opening price at the start of a trading day. It represents the initial price at which transactions occur when the market opens.

Close Price: The final traded price of a financial instrument at the end of a specific time period, such as the closing price at the end of a trading day. It is an important reference point for assessing the overall performance of the instrument during that period.

Volume: The total number of shares or contracts traded for a specific financial instrument during a given time period. Volume is a measure of market activity and liquidity. Higher volumes often indicate increased interest and participation in the market.

## Collecting Data

Download the historical forex exchange rate data from *yfinance*. Drop the 'Adj Close' column from each of the loaded dataframes. This column is commonly dropped when working with financial data if you want to focus on other features. Finally, remove any rows with missing values from each dataframe.

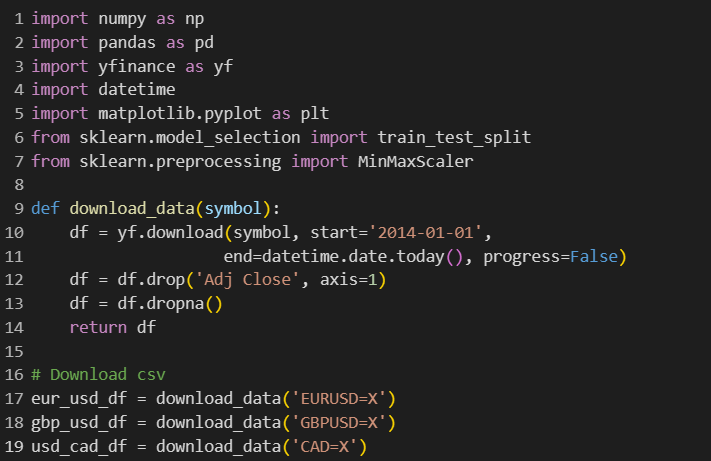


Figure 3.1: Download data

## Data Splitting

‘df\_to\_X\_y’ converts a given DataFrame into three distinct arrays: X, y, date. The features X will represent the current day, while the output column Y will correspond to the closing price of the following day.

A computer screen shot of text

Description automatically generated

Figure 3.2: Convert data to X, y function

A black background with white text

Description automatically generated

Figure 3.3: Convert data to X, y

Using MinMaxScaler from the scikit-learn library to perform feature scaling on three distinct datasets, namely eur\_usd\_X, gbp\_usd\_X, and usd\_cad\_X. In this context, the scaler is applied to each dataset individually, namely eur\_usd\_X, gbp\_usd\_X, and usd\_cad\_X, effectively normalizing the values within each dataset. This normalization is crucial for ensuring that the features across different datasets are on a comparable scale, preventing certain features from disproportionately influencing the machine learning models during training.

A screen shot of a computer program

Description automatically generated

Figure 3.4: Normalize the X dataset

Splitting three datasets into training and testing subsets for subsequent machine learning model training and evaluation. The datasets are initially scaled using the MinMaxScaler to normalize their values. Subsequently, the code determines the 80% quantile index (q80) for each dataset, effectively establishing the point at which the datasets are partitioned into an 80% training set and a 20% testing set.

A screenshot of a computer program

Description automatically generated

Figure 3.5: Split data into train and test

# CHAPTER 4. PREDICTING PRICE USING DEEP LEARNING MODELS

## Recurrent Neural Network Model

Create a sequential model with a SimpleRNN layer and a Dense output layer. Compile the model with the Adam optimizer and binary crossentropy loss for binary classification. Train the model on the generated training data for 10 epochs with a batch size of 32.

A computer screen with text and images

Description automatically generated

Figure 4.1: RNN Model

### EUR/USD Prediction Using RNN

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.2: RNN Prediction on EUR/USD Train data

A graph with orange and blue lines

Description automatically generated

Figure 4.3: Visualize the RNN prediction on EUR/USD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.4: RNN Prediction on EUR/USD Test data

A graph with numbers and lines

Description automatically generated

Figure 4.5: Visualize the RNN prediction on EUR/USD Test data

#### A graph showing a line of orange and red lines Description automatically generated with medium confidence

Figure 4.6: Visualize the RNN prediction on EUR/USD

### GBP/USD Prediction Using RNN

Train the model on gbp/usd training data using the fit method. Specify the number of epochs and batch size.



Figure 4.7: Train RNN model using GBP/USD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.8: RNN Prediction on GBP/USD Train data

A graph with numbers and lines

Description automatically generated

Figure 4.8: Visualize the RNN prediction on GBP/USD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.9: RNN Prediction on GBP/USD Test data

A graph of blue and orange lines

Description automatically generated

Figure 4.10: Visualize the RNN prediction on GBP/USD Test data

A graph of a stock market

Description automatically generated

Figure 4.11: Visualize the RNN prediction on GBP/USD

### USD/CAD Prediction Using RNN

Train the model on cad/usd training data using the fit method. Specify the number of epochs and batch size.

A black background with white text

Description automatically generated

Figure 4.12: Train RNN model using USD/CAD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.13: RNN Prediction on USD/CAD Train data

A graph with orange and blue lines

Description automatically generated

Figure 4.14: Visualize the RNN prediction on USD/CAD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.15: RNN Prediction on USD/CAD Test data

A graph with numbers and lines

Description automatically generated

Figure 4.16: Visualize the RNN prediction on USD/CAD Test data

A graph of a price

Description automatically generated with medium confidence

Figure 4.17: Visualize the RNN prediction on USD/CAD

## Long Short-Term Memory Model

Create a sequential model with a LSTM layer with 64 units and 2 Dense output layers. Compile the model with the Adam optimizer and Mean Squared Error loss function. Train the model on the generated training data for 10 epochs.

A computer screen shot of a program code

Description automatically generated

Figure 4.18: LSTM Model

### EUR/USD Prediction Using LSTM

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer program

Description automatically generated

Figure 4.19: LSTM Prediction on EUR/USD Train data

A graph of a stock market

Description automatically generated with medium confidence

Figure 4.20: Visualize the LSTM prediction on EUR/USD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.21: LSTM Prediction on Test data

A graph of a graph

Description automatically generated with medium confidence

Figure 4.22: Visualize the LSTM prediction on EUR/USD Test data

#### A graph of orange and blue lines Description automatically generated

Figure 4.23: Visualize the LSTM prediction on EUR/USD

### GBP/USD Prediction Using LSTM

Train the model on gbp/usd training data using the fit method. Specify the number of epochs and batch size.

#### 

Figure 4.24: Train LSTM model using GBP/USD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.25: LSTM Prediction on GBP/USD Train data

A graph with numbers and lines

Description automatically generated

Figure 4.26: Visualize the LSTM prediction on GBP/USD Train data

A computer screen with text

Description automatically generated

Figure 4.27: LSTM Prediction on GBP/USD Test data

A graph with blue and orange lines

Description automatically generated

Figure 4.28: Visualize the LSTM prediction on GBP/USD Test data

A graph of a stock market

Description automatically generated with medium confidence

Figure 4.29: Visualize the LSTM prediction on GBP/USD

### USD/CAD Prediction Using LSTM

Train the model on cad/usd training data using the fit method. Specify the number of epochs and batch size.



Figure 4.30: Train LSTM model using USD/CAD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A computer screen with text

Description automatically generated

Figure 4.31: LSTM Prediction on USD/CAD Train data

A graph of orange and blue lines

Description automatically generated

Figure 4.32: Visualize the LSTM prediction on USD/CAD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.33: LSTM Prediction on USD/CAD Test data

A graph with orange and blue lines

Description automatically generated

Figure 4.34: Visualize the LSTM prediction on USD/CAD Test data

A graph of orange and blue lines

Description automatically generated

Figure 4.35: Visualize the LSTM prediction onUSD/CAD

## Gated Recurrent Unit Model

Create a sequential model with 4 GRU layers are stacked on top of each other. Dropout layers are added after each GRU layer to prevent overfitting. The dropout rate decreases from 0.3 to 0.2 to 0.1, and then increases to 0.2 after the last GRU layer. A Dense layer with one unit is added as the output layer. It doesn't have an activation function, which implies it will output a continuous value. Compile the model with the Adam optimizer and Mean Squared Error loss function. Train the model on the generated training data for 10 epochs.

A screen shot of a computer program

Description automatically generated

Figure 4.36: GRU Model

### EUR/USD Prediction Using GRU

Make predictions using the trained model, and plot the prediction using Matplotlib.

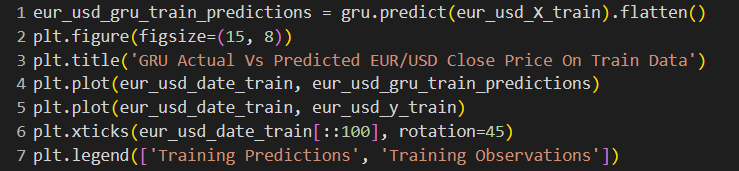


Figure 4.37: GRU Prediction on EUR/USD Train data

A graph of a graph

Description automatically generated with medium confidence

Figure 4.38: Visualize the GRU prediction on EUR/USD Train data

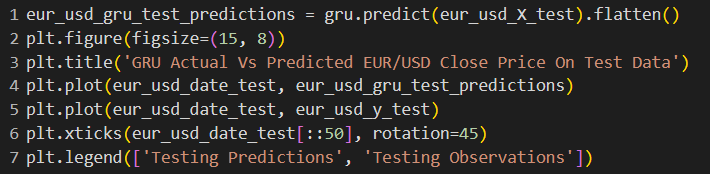


Figure 4.39: GRU Prediction on Test data

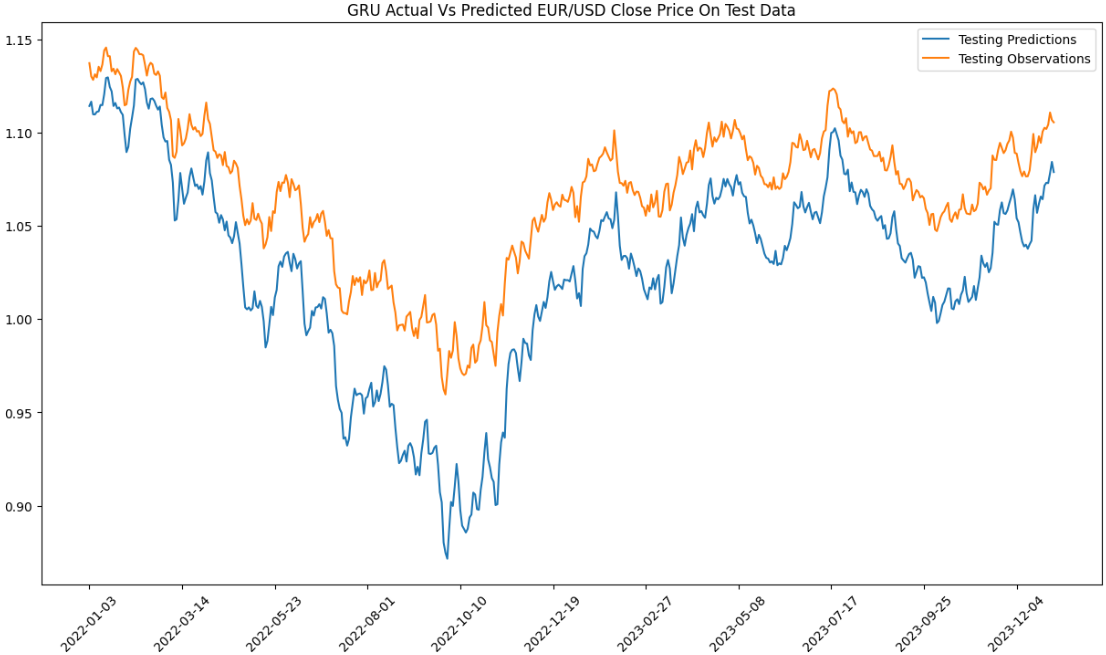


Figure 4.40: Visualize the GRU prediction on EUR/USD Test data

#### 

Figure 4.41: Visualize the GRU prediction on EUR/USD

### GBP/USD Prediction Using GRU

Train the model on gbp/usd training data using the fit method. Specify the number of epochs and batch size.

#### 

Figure 4.42: Train GRU model using GBP/USD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

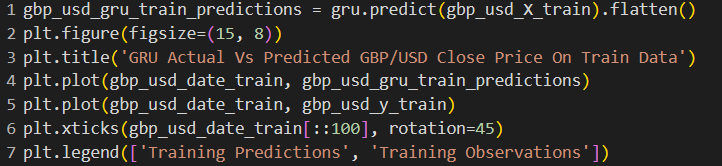


Figure 4.43: GRU Prediction on GBP/USD Train data

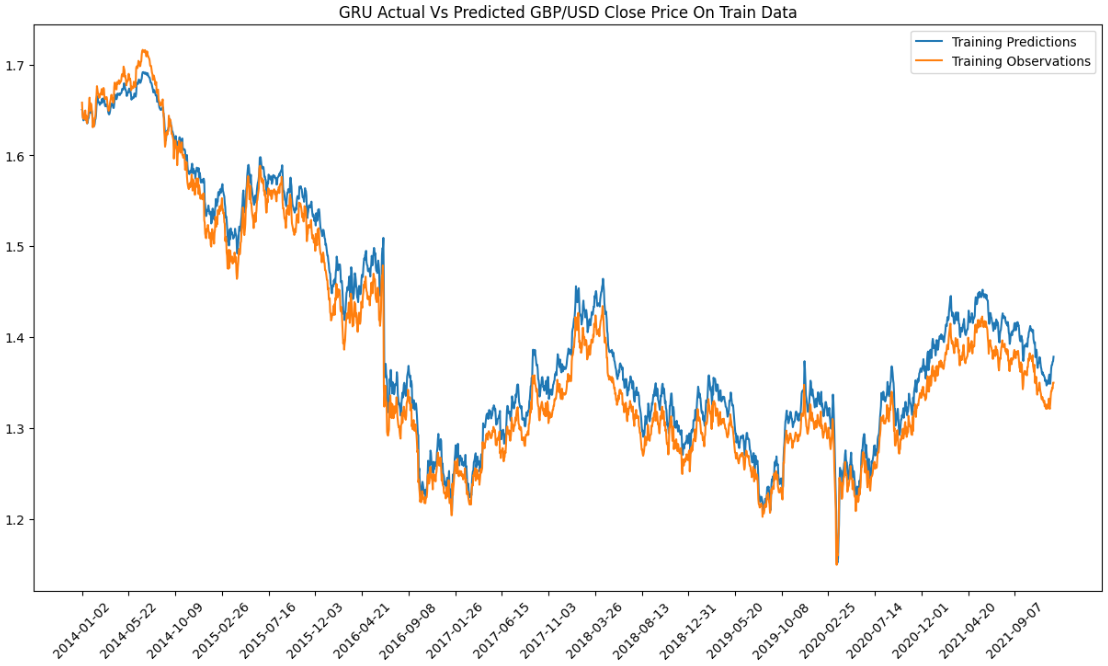


Figure 4.44: Visualize the GRU prediction on GBP/USD Train data

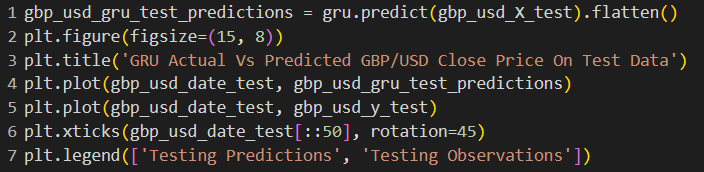


Figure 4.45: GRU Prediction on GBP/USD Test data

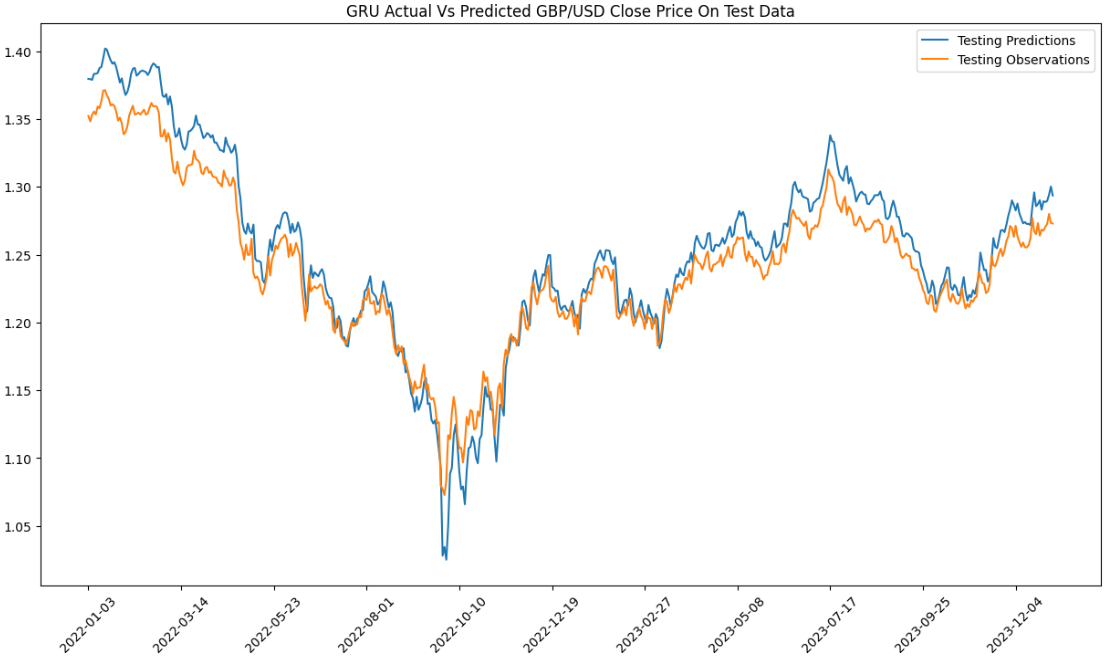


Figure 4.46: Visualize the GRU prediction on GBP/USD Test data

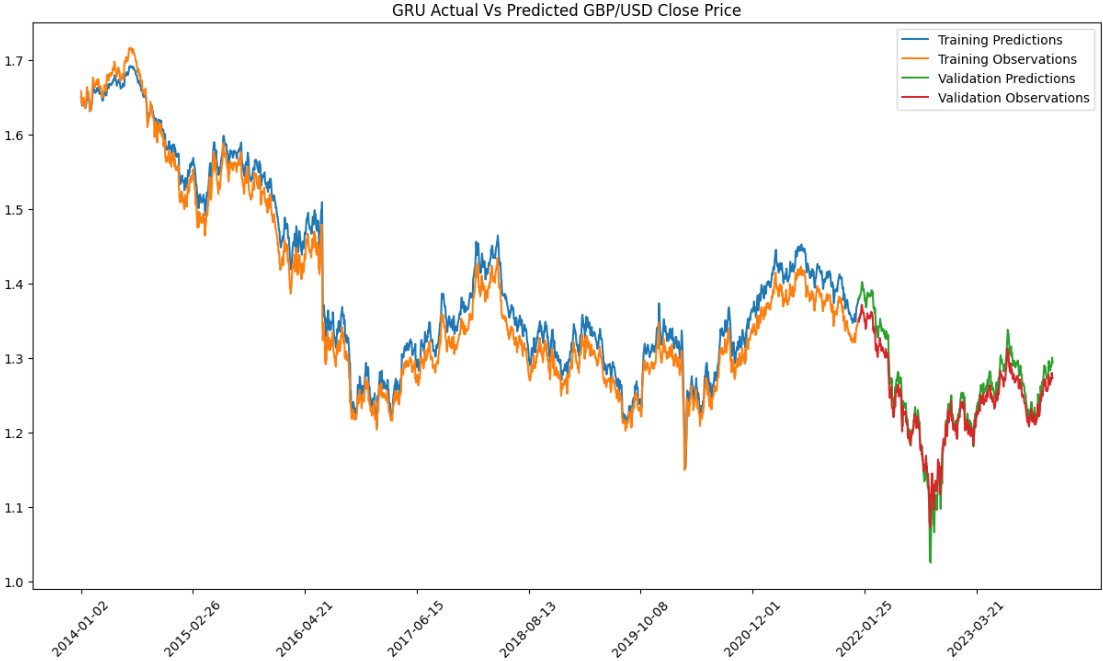


Figure 4.47: Visualize the GRU prediction on GBP/USD

### USD/CAD Prediction Using GRU

Train the model on cad/usd training data using the fit method. Specify the number of epochs and batch size.



Figure 4.48: Train GRU model using USD/CAD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

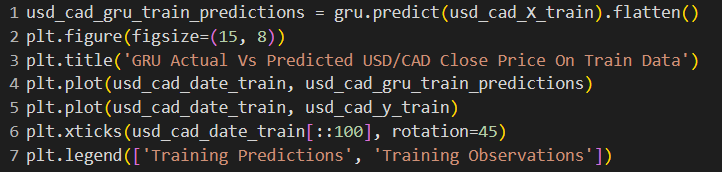


Figure 4.49: GRU Prediction on USD/CAD Train data

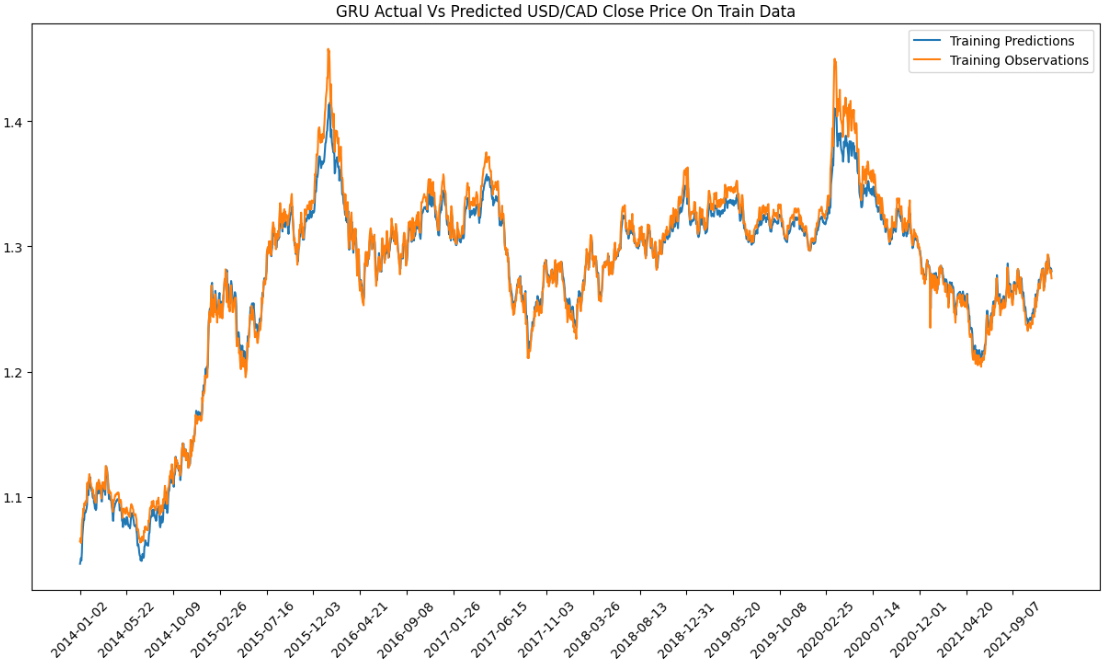


Figure 4.50: Visualize the GRU prediction on USD/CAD Train data

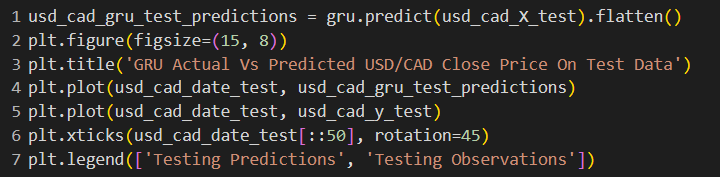


Figure 4.51: GRU Prediction on USD/CAD Test data

A graph of blue and orange lines

Description automatically generated

Figure 4.52: Visualize the GRU prediction on USD/CAD Test data

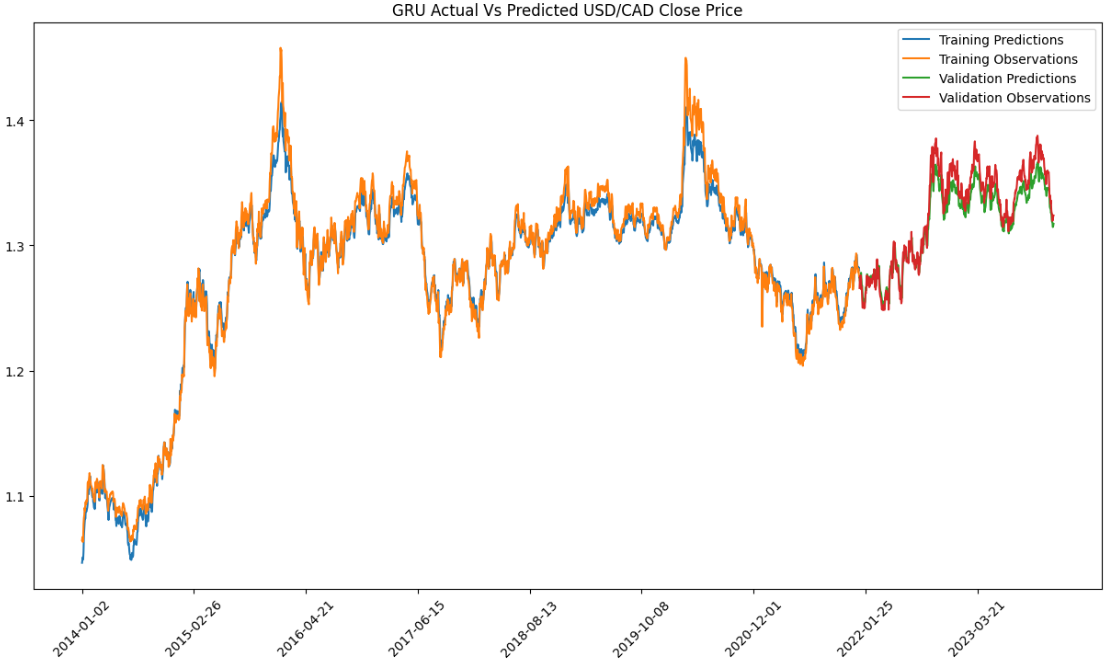


Figure 4.53: Visualize the GRU prediction on USD/CAD

## Artificial Neural Network Model

Create a sequential model with 2 Dense layers. The first Dense layer has 64 units, ReLU activation function. The second Dense layer has 32 units and a ReLU activation function. A Dropout layer with a dropout rate of 0.2 is added after the second Dense layer. Dropout is used to prevent overfitting by randomly setting a fraction of input units to zero during training. The model is compiled using the Adam optimizer and mean squared error as the loss function. Train the model on the generated training data for 10 epochs.

A screen shot of a computer program

Description automatically generated

Figure 4.54: ANN Model

### EUR/USD Prediction Using ANN

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.55: ANN Prediction on EUR/USD Train data

A graph with numbers and lines

Description automatically generated

Figure 4.56: Visualize the ANN prediction on EUR/USD Train data

A screen shot of a computer program

Description automatically generated

Figure 4.57: ANN Prediction on EUR/USD Test data

A graph with blue and orange lines

Description automatically generated

Figure 4.58: Visualize the ANN prediction on EUR/USD Test data

#### A graph with numbers and lines Description automatically generated

Figure 4.59: Visualize the ANN prediction on EUR/USD

### GBP/USD Prediction Using ANN

Train the model on gbp/usd training data using the fit method. Specify the number of epochs and batch size.

#### A black background with white text Description automatically generated

Figure 4.60: Train ANN model using GBP/USD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A computer screen with text

Description automatically generated

Figure 4.61: ANN Prediction on GBP/USD Train data

A graph with numbers and lines

Description automatically generatedFigure 4.62: Visualize the ANN prediction on GBP/USD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.45: ANN Prediction on GBP/USD Test data

A graph of blue and orange lines

Description automatically generated

Figure 4.46: Visualize the ANN prediction on GBP/USD Test data

A graph of a stock market

Description automatically generated with medium confidence

Figure 4.47: Visualize the ANN prediction on GBP/USD

### USD/CAD Prediction Using ANN

Train the model on cad/usd training data using the fit method. Specify the number of epochs and batch size.

A black background with white text

Description automatically generated

Figure 4.48: Train ANN model using USD/CAD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A computer screen with text

Description automatically generated

Figure 4.49: ANN Prediction on USD/CAD Train data

A graph of blue and orange lines

Description automatically generated

Figure 4.50: Visualize the ANN prediction on USD/CAD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.51: ANN Prediction on USD/CAD Test data

A graph of a graph

Description automatically generated with medium confidence

Figure 4.52: Visualize the ANN prediction on USD/CAD Test data

A graph of orange and blue lines

Description automatically generated

Figure 4.53: Visualize the ANN prediction on USD/CAD

## Multi-layer Perceptron Model

Create a sequential model with 2 Dense layers. The first Dense layer has 100 units with a ReLU activation function. The second Dense layer has 1 unit with a ReLU activation function. Compile the model with the Adam optimizer and Mean Squared Error loss function. Train the model on the generated training data for 10 epochs.

A screen shot of a computer program

Description automatically generated

Figure 4.54: MLP Model

### EUR/USD Prediction Using MLP

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.55: MLP Prediction on EUR/USD Train data

A graph with orange and blue lines

Description automatically generated

Figure 4.56: Visualize the MLP prediction on EUR/USD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.57: MLP Prediction on EUR/USD Test data

A graph of a stock market

Description automatically generated

Figure 4.58: Visualize the MLP prediction on EUR/USD Test data

#### A graph of a line graph Description automatically generated with medium confidence

Figure 4.59: Visualize the MLP prediction on EUR/USD

### GBP/USD Prediction Using MLP

Train the model on gbp/usd training data using the fit method. Specify the number of epochs and batch size.

A black background with white text

Description automatically generated

Figure 4.60: Train MLP model using GBP/USD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.61: MLP Prediction on GBP/USD Train data

A graph with numbers and lines

Description automatically generated

Figure 4.62: Visualize the MLP prediction on GBP/USD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.63: MLP Prediction on GBP/USD Test data

A graph with numbers and lines

Description automatically generated

Figure 4.64: Visualize the MLP prediction on GBP/USD Test data

A graph of orange and blue lines

Description automatically generated

Figure 4.65: Visualize the MLP prediction on GBP/USD

### USD/CAD Prediction Using MLP

Train the model on cad/usd training data using the fit method. Specify the number of epochs and batch size.



Figure 4.66: Train MLP model using USD/CAD data

Make predictions using the trained model, and plot the prediction using Matplotlib.

A screen shot of a computer code

Description automatically generated

Figure 4.67: MLP Prediction on USD/CAD Train data

A graph of orange lines

Description automatically generated

Figure 4.68: Visualize the MLP prediction on USD/CAD Train data

A screen shot of a computer code

Description automatically generated

Figure 4.69: MLP Prediction on USD/CAD Test data

A graph with numbers and lines

Description automatically generated

Figure 4.70: Visualize the MLP prediction on USD/CAD Test data

A graph of orange lines

Description automatically generated

Figure 4.71: Visualize the GRU prediction on USD/CAD

# CHAPTER 5. EXCHANGE RATE PREDICTION WEBSITE DEMO

## Setup

Write ‘model.py’ for defining models currency exchange rates using various time series predicting models, such as RNN, LSTM, GRU, MLP, and ANN. The script employs the download\_data function to retrieve historical currency exchange rate data using the Yahoo Finance API. Subsequently, the script generates visualizations for model accuracy, exchange rate predictions, and observed values on the test data, providing users with insights into the model's performance.

A screenshot of a computer program

Description automatically generated

Figure 5.1: model.py script

‘currency.py’ is designed to obtain predictions from a specified deep learning model and present pertinent information, including the predicted values for the day before the last day and the last day. The function employs the df\_to\_X\_y function to prepare the data for prediction. It then utilizes the provided model to generate predictions for the test data. The results, including the dates and corresponding predicted values for the day before the last day and the last day, are extracted and presented in a structured manner.

A screenshot of a computer program

Description automatically generated

Figure 5.2: currency.py script

‘app.py’ is the main application entry point for the Price Predictor App. It utilizes the Streamlit library to create a user interface and incorporates the predictive capabilities from the previously defined machine learning models in the ‘model.py’ module.

A screen shot of a computer

Description automatically generated

Figure 5.3: app.py script

Install Streamlit library and localtunnel package using npm. Run a Streamlit app (app.py) in the background and redirecting both standard output (stdout) and standard error (stderr) to a file named ‘logs.txt’.



Figure 5.4: Run Streamlit app

Retrieve the public IP address of the current machine using the "icanhazip.com" service. Additionally, it prints a message containing information about the potential use of this IP address as a password or endpoint for the localtunnel service.

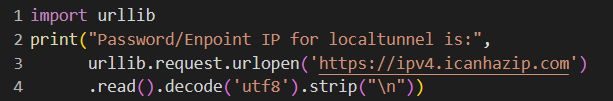


Figure 5.5: Retrieve Password/Enpoint IP

Run localtunnel with provided password/enpoint IP and expose a local web server running on port 8501 to the internet.



Figure 5.6: Run localtunnel

## Demo

Users can choose from 3 major currency pairs to tailor predictions to their specific needs.

A screenshot of a computer

Description automatically generated

Figure 5.*7*: Select pairs of currency

Users have the flexibility to choose from a selection of five distinct models for their prediction purposes.

A screenshot of a computer

Description automatically generated

Figure 5.8: Select models

Subsequently, users are required to input the closing, opening, high, and low prices within specified fields for prediction.

A screenshot of a black screen

Description automatically generated

Figure 5.9: Input prices

Upon the completion of the selection process for currency pairs and models, users are required to click the "Accept" button to initiate the calculation and visualization procedures.

A screenshot of a computer

Description automatically generated

Figure 5.10: Begin visualization

Following a brief interval, the website will present comprehensive information, encompassing model accuracy metrics, the prediction for the input prices and a graphical representation illustrating the observed and predicted price trends leading up to the present moment. This structured presentation aims to provide users with a detailed overview of the predicting accuracy and the temporal alignment of predicted values in relation to the observed data, facilitating a thorough analysis of the exchange rate predictions.

A screen shot of a graph

Description automatically generated

Figure 5.11: Display prediction and visualization

# CHAPTER 6: CONCLUSION

## Conclusion

The application of deep learning models for exchange rate predicting represents a promising avenue for enhancing predictive accuracy and capturing complex patterns within financial time series data. Throughout this report, we explored the implementation and evaluation of various deep learning architectures, including Recurrent Neural Networks, Long Short-Term Memory, Gated Recurrent Unit, Multi-Layer Perceptron, and Artificial Neural Network.

The models exhibited varying degrees of performance, with each demonstrating strengths in specific scenarios. The RNN, LSTM, and GRU models, designed to capture temporal dependencies, showcased notable accuracy in predicting exchange rate movements over time. The MLP and ANN models, emphasizing feedforward architectures, proved effective in capturing non-linear relationships within the data.

The interactive web application, developed for users to customize predictions based on currency pairs and models of their choice, provides a user-friendly interface to explore the capabilities of these deep learning models. The incorporation of features such as model accuracy metrics, historical prediction visualization, and dynamic model selection enhances the application's utility for both novice and experienced users.

In conclusion, the amalgamation of deep learning models with an interactive and accessible web interface not only empowers users to make informed decisions but also fosters a deeper understanding of the dynamic relationship between financial markets and predictive analytics. The pursuit of innovation in model architectures and a commitment to adaptability will undoubtedly contribute to the continuous evolution of exchange rate predicting in the realm of deep learning.

## Development Direction

Ensemble Models: Consider building ensemble models that combine predictions from multiple deep learning models. Ensemble techniques often lead to more robust and accurate prediction by leveraging the strengths of different models.

Explainability and Interpretability: Enhance the interpretability of the models by incorporating techniques that provide insights into how the models arrive at their predictions. This is particularly important in finance for building trust and understanding.

Evaluation Metrics: Evaluate models using a variety of metrics beyond accuracy, such as precision, recall or F1 score. Different metrics may be more relevant depending on the specific objectives of the predicting task.

Future Research Directions: Discuss potential avenues for future research in exchange rate predicting. Identify emerging trends, technologies, or methodologies that could further advance the field.

# REFERENCES

aakarshachug. (2023). *Deep Learning | Introduction to Long Short Term Memory*. From geeksforgeeks.org: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

aishwarya.27. (2023). *Introduction to Recurrent Neural Network*. From geeksforgeeks.org: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

Banoula, M. (2023). *An Overview on Multilayer Perceptron (MLP)*. From simplilearn.com: https://www.simplilearn.com/tutorials/deep-learning-tutorial/multilayer-perceptron

GANTI, A. (2023). *Foreign Exchange Market: How It Works, History, and Pros and Cons*. From investopedia.com: https://www.investopedia.com/terms/forex/f/foreign-exchange-markets.asp

Saxena, S. (2023). *Introduction to Gated Recurrent Unit (GRU)*. From analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/

Singh, G. (2024). *Introduction to Artificial Neural Networks*. From analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2021/09/introduction-to-artificial-neural-networks/