

# **KANTIPUR ENGINEERING COLLEGE**

**(Affiliated to Tribhuvan University)**

**Dhapakhel, Lalitpur**



**[Subject Code: CT755]**

## **A MAJOR PROJECT PROPOSAL ON STOCK PRICE PREDICTION**

**Submitted by:**

**Aman Devkota [KAN076BCT010]**

**Ankur Karmacharya [KAN076BCT013]**

**Prashad Adhikary [KAN076BCT056]**

**A MAJOR PROJECT SUBMITTED IN PARTIAL  
FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE  
OF BACHELOR IN COMPUTER ENGINEERING**

**Submitted to:**

**Department of Computer and Electronics Engineering**

**June, 2023**

# **STOCK PRICE PREDICTION**

**Submitted by:**

**Aman Devkota [KAN076BCT010]**

**Ankur Karmacharya [KAN076BCT013]**

**Prashad Adhikary [KAN076BCT056]**

**A MAJOR PROJECT SUBMITTED IN PARTIAL  
FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE  
OF BACHELOR IN COMPUTER ENGINEERING**

**Submitted to:**

**Department of Computer and Electronics Engineering**

**Kantipur Engineering College**

**Dhapakhel, Lalitpur**

**June, 2023**

## ABSTRACT

Stock market prediction is when people try to figure out what the value of a stock will be in the future. They do this to make money by buying and selling stocks at the right time. Deep learning models are used to help predict stock prices. Recurrent neural networks are a type of deep learning model that is often used. There are different types of deep learning models that can be used depending on the situation. Predicting stock prices is difficult because there are many factors that can affect them. These factors can include things like politics, global economic conditions, and a company's financial performance. This project performs a comparative analysis of three deep learning models-the Long Short-term Memory (LSTM), Gated Recurrent Unit (GRU), and Vanilla RNN (VRNN)- in predicting the next day's closing price of the Nepal Stock Exchange (NEPSE) index. The performances of employed models are compared using the standard assessment metric- Mean Absolute Percentage Error (MAPE).

**Keywords**— *LSTM, GRU, VRNN, NEPSE, Deep learning*

# TABLE OF CONTENTS

<b>Abstract</b>	<b>i</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Abbreviations</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement . . . . .	1
1.3 Objectives . . . . .	2
1.4 Project Features . . . . .	2
1.5 Application Scope . . . . .	2
1.6 System Requirement . . . . .	3
1.6.1 Development Requirements . . . . .	3
1.6.1.1 Software Requirements . . . . .	3
1.6.1.2 Hardware Requirements . . . . .	3
1.6.2 Deployment Requirements . . . . .	3
1.6.2.1 Software Requirements . . . . .	3
1.6.2.2 Hardware Requirements . . . . .	3
1.7 Project Feasibility . . . . .	3
1.7.1 Technical Feasibility . . . . .	3
1.7.2 Operational Feasibility . . . . .	4
1.7.3 Economic Feasibility . . . . .	4
1.7.4 Schedule Feasibility . . . . .	4
<b>2 Literature Review</b>	<b>5</b>
2.1 Related Projects . . . . .	5
2.1.1 AI Stock Finder . . . . .	5
2.1.2 Wallstreetzen Stock Forecast . . . . .	5
2.2 Related Works . . . . .	5
<b>3 Methodology</b>	<b>6</b>
3.1 Working Mechanism . . . . .	6
3.1.1 Data set . . . . .	6
3.1.2 Split the data set into train set and test set . . . . .	8
3.1.3 Data Preprocessing . . . . .	9

3.1.4	Train the model . . . . .	9
3.1.4.1	Long Short term Memory (LSTM) . . . . .	9
3.1.4.2	Vanilla RNN (VRNN) . . . . .	10
3.1.4.3	Gated Recurrent Unit Network . . . . .	11
3.1.5	Predict the stock price . . . . .	12
3.1.6	Check for accuracy . . . . .	12
3.2	System Diagram . . . . .	13
3.2.1	Use case diagram . . . . .	13
3.2.2	Software Development Model . . . . .	14
<b>4</b>	<b>Epilogue</b>	<b>15</b>
4.1	Expected Output . . . . .	15
	<b>References</b>	<b>15</b>

## LIST OF FIGURES

1.1	Gantt Chart . . . . .	4
3.1	Working mechanism of Stock Price Prediction System . . . . .	6
3.2	LSTM . . . . .	10
3.3	VRNN . . . . .	11
3.4	GRU network . . . . .	12
3.5	Use case Diagram of Stock Price Prediction System . . . . .	13
3.6	Incremental Model . . . . .	14

## **LIST OF ABBREVIATIONS**

ATR: Average Time Range

CBIR: Commercial Bank Interest Rate

EMA: Exponential Moving Average

ER: Exchange Rate

GRU: Gated Recurrent Unit

IR: Inflation Rate

LSTM: Long Short Term Memory

MACD: Moving Average Convergence Divergence

MAPE: Mean Absolute Percentage Error

MFI: Money Flow Index

NEPSE: Nepal Stock Exchange

RSI: Relative Strength Index

TRB: Treasury Bills

VRNN: Vanilla Recurrent Neural Network

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 Background**

The stock price is determined by the highest price a buyer is willing to pay or the lowest price a seller is willing to accept. Supply and demand are key factors that can affect stock prices. High demand can lead to an increase in stock price, while high supply can lead to a decrease. However, it is difficult to determine the exact factors that contribute to changes in demand and supply. Stock market prediction involves forecasting the future value of a stock. Financial analysts use two main schools of thought for analyzing and predicting stock markets: technical analysis and fundamental analysis. [1]

Machine learning models can learn a function by analyzing data without explicit programming. The performance of these algorithms depends on the representation of the data. However, stock market time-series data is difficult to map and is best described as a random walk, making feature engineering and prediction challenging. Therefore, deep learning models are the best available tool for stock market prediction.[2]

There are two schools of thought in developing predictive models for estimating stock prices. Classical thinking uses historical facts and indicators to predict future stock prices, often through variations of Autoregressive Integrated Moving Average (ARIMA) models. However, these statistical models may not efficiently capture the noisy and non-linear behavior of stock market data. Modern theory assumes that historical data cannot reflect the exact upcoming structure due to inherent non-linearity in stock market data. With the rapid advancement of artificial intelligence and machine learning techniques, availability of large-scale data, and increased computational capabilities, robust machine learning models can capture nonlinear behavior and predict stock prices.[3]

### **1.2 Problem Statement**

The stock market is a frequent topic in the media, with news outlets reporting on its daily fluctuations. Whenever the market experiences a new peak or dip, it garners significant attention. Developing an effective algorithm to forecast short-term stock prices could



potentially boost investment rates and create more business prospects in the market.

### **1.3 Objectives**

The primary objectives of this projects are as follows:

- i. Gathering the multifaceted information of NEPSE index, putting them together into a common framework, and constructing a reliable model for accurate predictions.
- ii. Conducting extensive data driven experimentation using customized parameters of LSTM, GRU, and VRNN models.
- iii. Performing comparative study of deep learning models (LSTM, GRU, and VRNN) for the best fit and forecasting under the identical conditions.
- iv. Conducting statistical experiment to validate and verify the reliability and robustness of the model.

### **1.4 Project Features**

The project will be able to accomplish following:

- Reliable
- Accurate
- User friendly
- Efficient

### **1.5 Application Scope**

Stock price prediction has various applications such as investment decision-making, risk management, algorithmic trading, portfolio optimization, and economic forecasting. Predicting stock prices can help investors make informed decisions, manage risk, optimize portfolios, and provide insights into the economy. It is a valuable tool for investors, financial institutions, and economists seeking to make informed decisions and manage risk.

## **1.6 System Requirement**

### **1.6.1 Development Requirements**

#### **1.6.1.1 Software Requirements**

- Windows/Linux/Mac
- HTML/CSS/JS
- Jupyter Notebook
- Python IDE

#### **1.6.1.2 Hardware Requirements**

- PC with at least 4-8 GB RAM
- Higher graphics of at least 2 GB

### **1.6.2 Deployment Requirements**

#### **1.6.2.1 Software Requirements**

- Web browser
- Visual studio code
- Pycharm

#### **1.6.2.2 Hardware Requirements**

- More than 1.5 GHz clock speed
- Minimum 4 GB RAM

## **1.7 Project Feasibility**

### **1.7.1 Technical Feasibility**

The technical feasibility assessment is focused on gaining in understanding of the present technical resources required by the system and their applicability to the expected needs of the proposed system. Regarding the proposed system, the technical requirement includes a PC.

### 1.7.2 Operational Feasibility

The user will not need any formal knowledge about programming so our project is operationally feasible.

### 1.7.3 Economic Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the user that the proposed system will provide. Most of the software used for the development is free. Thus, the project is economically feasible.

### 1.7.4 Schedule Feasibility

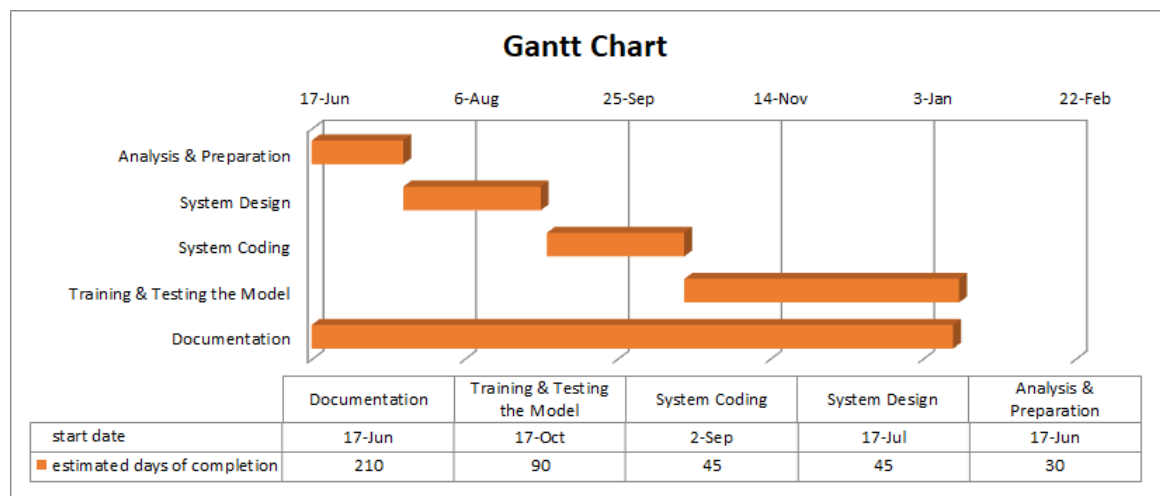


Figure 1.1: Gantt Chart

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Related Projects**

##### **2.1.1 AI Stock Finder**

It is a website that uses artificial intelligence algorithm to produce its forecasts based on 140 variables to yield high quality buy and sell signals.

##### **2.1.2 Wallstreetzen Stock Forecast**

WallStreetZen provides stock predictions and forecasts for the next 5 years. Their platform offers stock market forecasts for 2023, 2024, 2025, 2026, and 2027, along with analyst forecasts, price targets, buy/sell ratings, revenue/earnings forecasts, and more. Additionally, WallStreetZen offers a stock screener for identifying high-growth stocks based on analyst projections for the next 3 years . While it is important to note that stock market predictions and forecasts should be taken with caution, WallStreetZen provides a valuable tool for investors seeking insights into potential investment opportunities and market.

#### **2.2 Related Works**

Arjun Singh Saud and Subarna Shakya researched in detail about stock price prediction using different deep learning models. In the paper “Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE” published on 2020, there is comparison of the predictions by the deep learning models used in the study: LSTM, VRNN and GRU. [1] Arjun Singh Saud and Subarna Shakya again in their paper ”Analysis of Gradient Descent Optimization Techniques With Gated Recurrent Unit For Stock Price Prediction: A Case study on Banking Sector of Nepal Stock Exchange”, have compared stock price prediction performance of GRU with three widely used gradient descent optimization techniques: Momentum, RMSProp, and Adam.[2] Nawa Raj Pokhrel, Keshab Raj Dahal, Ramchandra Rimal, Hum Nath Bhandari, Rajendra K.C Khatri, Binod Rimal and William Edward Hahn on their paper ” Predicting NEPSE price index using deep learning models”, have performed comparative analysis between the deep learning models on the basis of assessment metrics like Root Mean Square Error, Mean Absolute Percentage Error and Correlation Coefficient. [3]

## CHAPTER 3 METHODOLOGY

### 3.1 Working Mechanism

The development of stock price prediction system involves major steps which is depicted in the diagram given below:

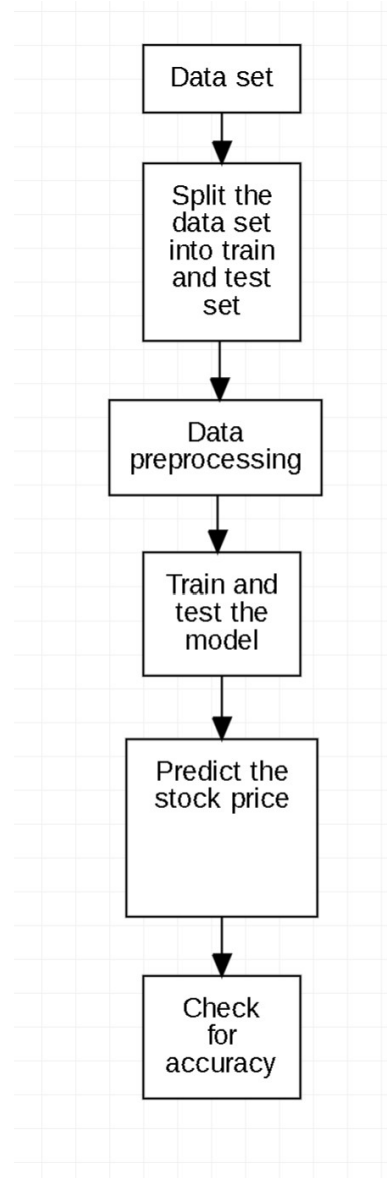


Figure 3.1: Working mechanism of Stock Price Prediction System

#### 3.1.1 Data set

The data set contains financial data, macroeconomic data and technical indicators. Fundamental data is significant or historical data that provides important information

needed for stock trading. It has an opening price, closing price, high price, low price and volume. The opening price is the first trading price at the opening of the trading day and the closing price is the last trading price of the stock that day. Likewise, the high and low prices are the highest and lowest prices for that day. Volume refers to the number of products traded in a trading day and indicates traders' interest in the market. High volume means more satisfaction and repetition. All business history data is daily data entry from Share Sansar portal, is one of the popular products providing detailed information about Nepal stock market.

Macroeconomic data is a macroeconomic variable that affects stock market performance. Under the umbrella of macroeconomic factors, the representative features that affect stock price forecasts are remittances (RMT), inflation rate (IR), commercial bank interest rates (CBIR), treasury bills (TRB), consumer price index (CPI) and exchange rate (ER). Remittance is the money Nepalese workers send from abroad. Inflation is the rate of increase in prices over a given period of time. The commercial bank interest rate is the bank rate at which Nepal Rastra Bank lends money to domestic financial institutions. The treasury bill is typically a promissory maturity note issued by a government as a primary instrument for regulating money supply and raising funds via open market operations. The Consumer Price Index is a measure of the average change overtime in the prices paid by urban consumers for a market basket of consumer goods and services. Exchange rate is the value of one country's currency against another country's currency.

Technical indicator includes Moving Average Convergence Divergence (MACD), Average True Range (ATR), Relative Strength Index (RSI), and Money Flow Index (MFI). MACD is calculated by subtracting the 26-day exponential moving average (EMA) from the 12-day EMA. ATR measures the market volatility and is defined as follows:

$$TrueRange(TR) = \max(H - L, |H - C_p|, |L - C_p|) \quad (3.1)$$

$$AverageTrueRange(TR) = \frac{1}{n} \sum_i^n TR_i \quad (3.2)$$

where,  $H, L, C_p$  represent current high, current low, previous close prices and  $TR_i, n$  represent a particular true range, the time period respectively.

The RSI is a momentum indicator that signals whether a security is overbought or oversold with current price levels, which is computed as follows:

$$RSI = 100 - \frac{100}{1 + \frac{AverageGain}{AverageLoss}} \quad (3.3)$$

where average gain and average loss are the average percentage gain and loss calculated over the certain look-back period.

The MFI communicates a possible reversal in time and provides a signal for further investment. Its calculation starts by finding the typical price (TP), the average of high, low, and close prices for each trading day. If the current TP is higher than the previous one, then positive money flow is calculated by multiplying the current TP with its volume. Similarly, a negative money flow is obtained if the current typical price is lower than the previous one. If the TP does not change, both positive and negative money flow will be zero. Summing all positive money flow indexes leads to positive money flow for a particular period. Negative money flow is calculated similarly. Mathematically, MFI is defined as:

$$MFI = 100 - \frac{100}{1 + MoneyRatio} \quad (3.4)$$

### 3.1.2 Split the data set into train set and test set

The data set is divided in train set and test set so that the train set can be used for training the model.

### 3.1.3 Data Preprocessing

Data preprocessing consists of three major steps: Data cleaning, Feature selection and Time series analysis. Data cleaning includes removing or handling missing values, outliers, and errors in the data. This may involve techniques like imputation, interpolation, or removing problematic data points. Identifying the relevant features or variables that may affect stock prices. This could include factors like historical prices, trading volumes, technical indicators (e.g., moving averages, RSI), fundamental data (e.g., earnings, revenue), or macroeconomic indicators (e.g., interest rates, GDP) comes under Feature selection and Time series analysis consists of Techniques such as identifying trends, seasonality, or auto correlation can be applied to analyze and preprocess the data as stock price data is often represented as a time series.

### 3.1.4 Train the model

The models used in this project for stock price prediction are: Vanilla RNN (VRNN), Long Shortterm Memory (LSTM), and Gated Recurrent Unit (GRU).

#### 3.1.4.1 Long Short term Memory (LSTM)

LSTM is a popular deep learning technique in RNN for time series prediction. While standard RNNs outperform traditional networks in preserving information, they are not very effective in learning long term dependencies due to the vanishing gradient problem. An LSTM is well-suited to classify and/or predict time-series data. There are several architectures of LSTM units. A common architecture is composed of a memory cell, an input gate, an output gate and a forget gate. The mathematical formulation of the LSTM cell is given below:

$$f_t = \sigma(x_t W_f + H_{t-1} U_f) \quad (3.5)$$

$$o_t = \sigma(x_t W_o + H_{t-1} U_o) \quad (3.6)$$

$$S_t = \sigma(S_{t-1} * f_t + i_t * H'_t) \quad (3.7)$$

$$i_t = \sigma(x_t W_i + H_{t-1} U_i) \quad (3.8)$$

$$H'_t = \tanh(x_t W_g + H_{t-1} U_g) \quad (3.9)$$

$$H_t = \tanh(S_t) * o_t \quad (3.10)$$



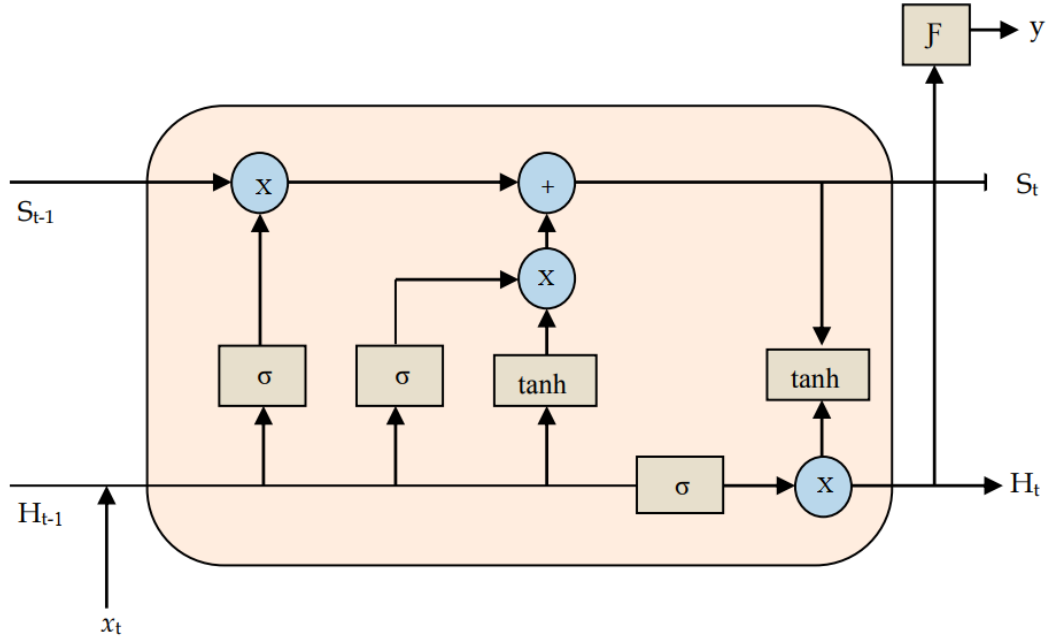


Figure 3.2: LSTM

#### 3.1.4.2 Vanilla RNN (VRNN)

It is the most basic form of a recurrent neural network. The simplest form of VRNN can be represented as in the figure given below. Normally, the tanh activation function is used in the hidden recurrent layer and the activation function for the output layer is selected according to the need of the problem to be solved. The operation of vanilla RNN can be expressed mathematically as below:

$$H_t = f(W_{xh}X_t + W_{hh}H_{t-1}) \quad (3.11)$$

$$O_t = g(W_{xo}S_t) \quad (3.12)$$

where  $X_t$  is input at time  $t$ ,  $H_t$  is state info at time  $t$ ,  $W_{xh}$ ,  $W_{hh}$  and  $W_{xo}$  are weight matrices.

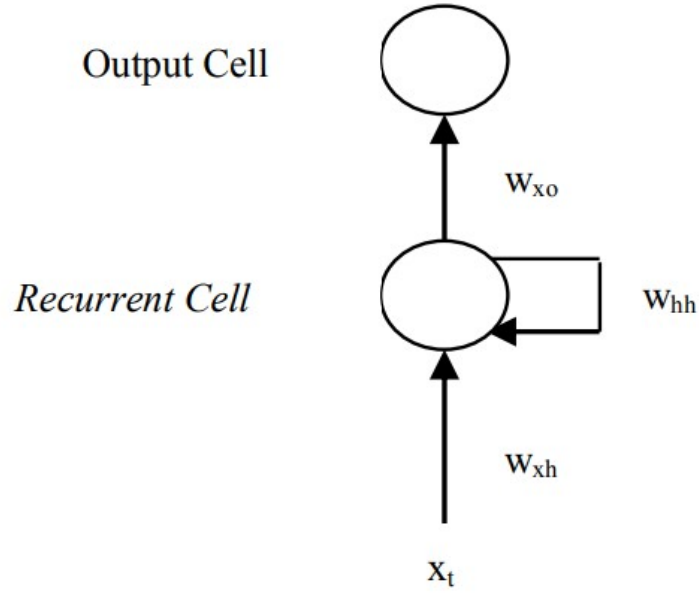


Figure 3.3: VRNN

#### 3.1.4.3 Gated Recurrent Unit Network

GRU is also a type of RNN architecture and aims to solve the vanishing gradient problem. GRU is similar to the LSTM but simplified in structure. GRU uses two gates: update gate and reset gate. The update gate of GRU is equivalent to the forget gate and input gate of LSTM. GRU has fewer parameters and thus may train faster. GRU cell takes two different pieces of information: the current input sequence  $x_t$ , the short-term memory from the previous cell  $H_{t-1}$ , at time  $t$ . The update gate carries the long-term dependencies in GRU. It determines the past information that needs to be passed into the next step. The reset gate takes the information from  $x_t$  and  $H_{t-1}$  and produces the output between 0 and 1 through the sigmoid layer and then it identifies which information to discard from the previous hidden state  $H_{t-1}$ . When the value is 1, it stores all the information in the cell while with a value of 0, it forgets all the information from the previous hidden state. Based on empirical evidence, LSTM and GRU have proven their effectiveness on many machine learning tasks. The operation of GRU network can be

expressed as follows:

$$Z_t = \sigma(W_z x_t + U_z H_{t-1}) \quad (3.13)$$

$$H'_t = \tanh(W_h x_t + (R_t * H_{t-1}) U_h) \quad (3.14)$$

$$R_t = \sigma(W_r x_t + U_r H_{t-1}) \quad (3.15)$$

$$H_t = (Z_t * H'_t) + ((1 - Z_t) * H_{t-1}) \quad (3.16)$$

where  $Z$  and  $R$  are update and reset gates, and  $H'_t$  and  $H_t$  are candidate hidden state and hidden states.

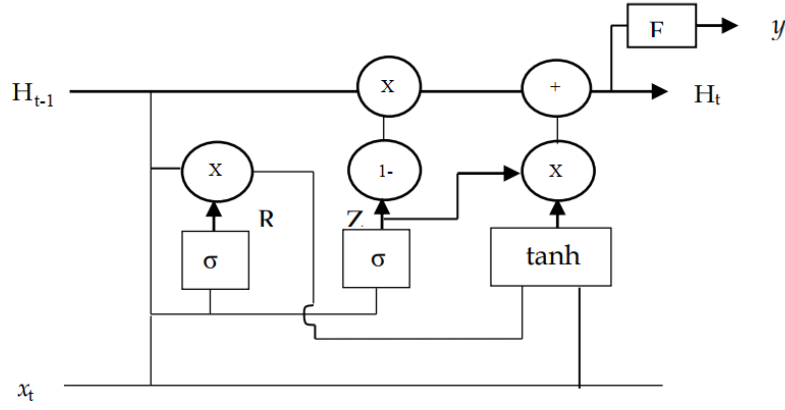


Figure 3.4: GRU network

### 3.1.5 Predict the stock price

Once the model is trained and evaluated, it can be used to make predictions on new, unseen data. This involves feeding the input features of the new data into the trained model, which then generates predictions for the corresponding stock prices.

### 3.1.6 Check for accuracy

Mean Absolute Percentage Error (MAPE) is a measure of prediction accuracy in statistics. It is calculated as the average of the unsigned percentage error, which is the absolute value of the difference between the actual value and the forecast value divided by the actual value, multiplied by 100. The formula for Mean Absolute Percentage Error

is given as:

$$M = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (3.17)$$

## 3.2 System Diagram

### 3.2.1 Use case diagram

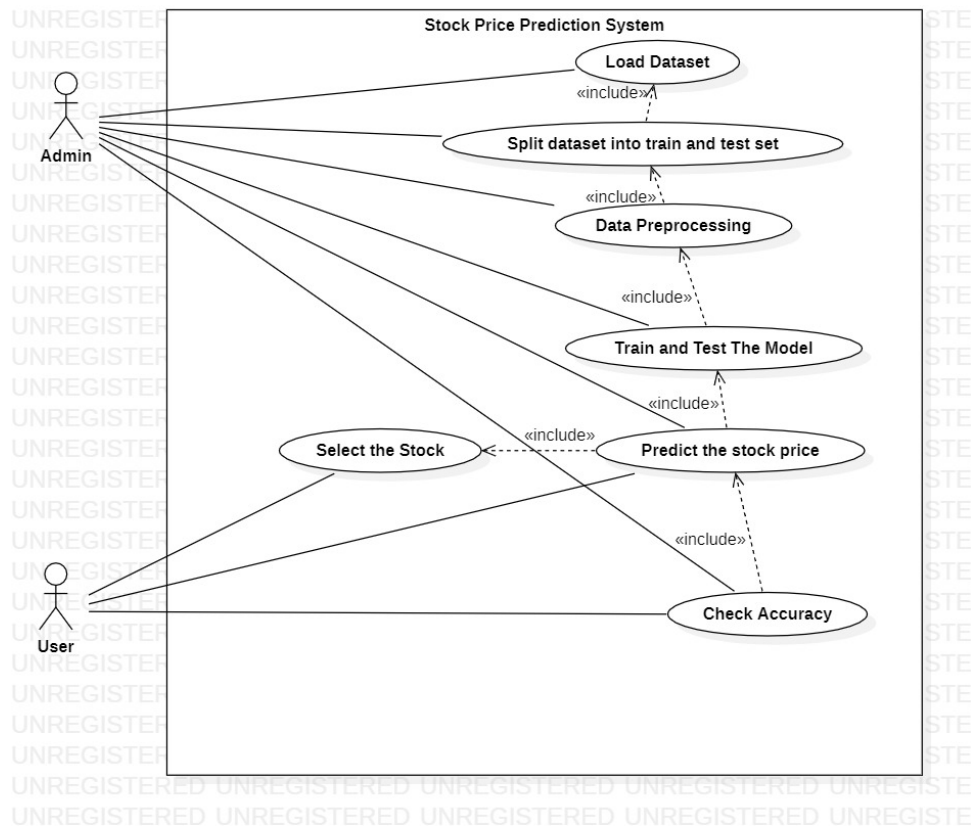


Figure 3.5: Use case Diagram of Stock Price Prediction System

### 3.2.2 Software Development Model

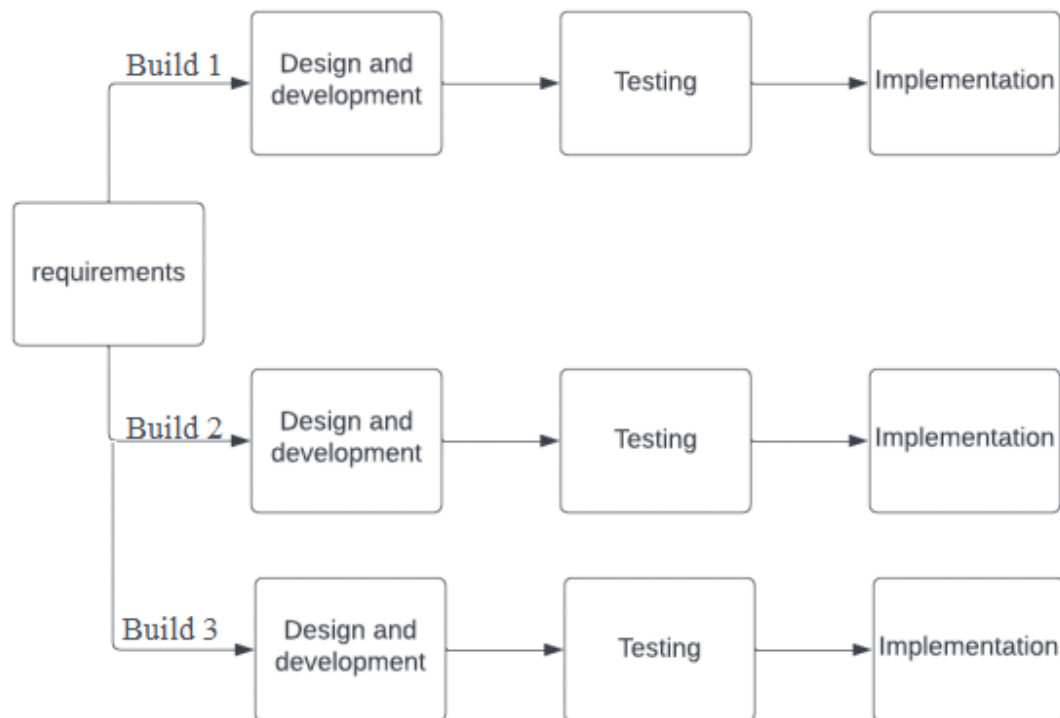


Figure 3.6: Incremental Model

Incremental model is a method of software engineering that combines the elements of waterfall model in iterative manner. It involves both development and maintenance. In this model requirements are broken down into multiple modules. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. Each iteration passes through the requirements, design, coding and testing phases. The first increment is often a core product where the necessary requirements are addressed, and the extra features are added in the next increments. The core product is delivered to the client. Once the core product is analyzed by the client, there is plan development for the next increment.

## **CHAPTER 4**

### **EPILOGUE**

#### **4.1 Expected Output**

Upon the completion of the project, the system will be able to predict the price of the stock the user selects.

## REFERENCES

- [1] A. S. Saud and S. Shakya, “Analysis of look back period for stock price prediction with rnn variants: A case study on banking sector of nepse,” *Procedia Computer Science*, vol. 167, pp. 788–798, 2020.
- [2] —, “Analysis of gradient descent optimization techniques with gated recurrent unit for stock price prediction: A case study on banking sector of nepal stock exchange,” *Journal of Institute of Science and Technology*, vol. 24, no. 2, pp. 17–21, 2019.
- [3] N. R. Pokhrel, K. R. Dahal, R. Rimal, H. N. Bhandari, R. K. Khatri, B. Rimal, and W. E. Hahn, “Predicting nepse index price using deep learning models,” *Machine Learning with Applications*, vol. 9, p. 100385, 2022.