UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Research and Development

**BACHELOR THESIS**

By

**Nguyen Anh Duy – BI12-127**

Data Science

Title:

**“Research and implementation of deep learning methods in stock price analysis and forecasting”**

External supervisor: Nguyen Duc Hoan

Internal supervisor: Nghiem Thi Phuong

**Hanoi, 2024**



Abstract

Acknowledgement

Contents

[Chapter 1: Introduction 1.1 Problem Statement 6](#_Toc170850549)

[1.2 Project Scope 6](#_Toc170850550)

[Objectives 6](#_Toc170850551)

[Research Subjects and Scope 7](#_Toc170850552)

[Methods 7](#_Toc170850553)

[1.3 Thesis Structure 7](#_Toc170850554)

[Chapter 2: Methodology 8](#_Toc170850555)

[2.1 Data Collection 8](#_Toc170850556)

[2.2 Data Preprocessing 10](#_Toc170850557)

[2.3 Data Mining 11](#_Toc170850558)

[2.4 ARIMA Model 16](#_Toc170850559)

[2.5 LSTM Model **The architecture of LSTM Networks** 16](#_Toc170850560)

[2.5 Comparison and Evaluation 18](#_Toc170850561)

[2.6 Tools and Technologies 18](#_Toc170850562)

[Chapter 3: Results and Discussion 18](#_Toc170850563)

[3.1 ARIMA Model Results 18](#_Toc170850564)

[3.2 LSTM Model Results 23](#_Toc170850565)

[3.3 Comparison of ARIMA and LSTM 28](#_Toc170850566)

[3.4 Impact of Real-Time Data Integration 28](#_Toc170850567)

[3.5 Discussion 28](#_Toc170850568)

[Chapter 4: Conclusion and Future Work 28](#_Toc170850569)

[4.1 Summary of Findings 28](#_Toc170850570)

[4.2 Implications 29](#_Toc170850571)

[4.3 Limitations 29](#_Toc170850572)

[4.4 Future Work 29](#_Toc170850573)

[References 29](#_Toc170850574)

[Appendices 29](#_Toc170850575)

List of Abbreviations

List of Figures

- Introduction to the topic  
- Objectives of the research  
- Methods used (LSTM, ARIMA, Data Crawling)  
- Key findings  
- Conclusions and implications

# Chapter 1: Introduction 1.1 Problem Statement

In the context of increasingly volatile and complex financial markets, the demand for accurate stock price prediction methods has become more urgent. Investors and market analysts continuously seek advanced tools and models to optimize investment decisions and mitigate risks. This project employs two popular time-series forecasting models, ARIMA and LSTM, to predict stock prices in response to this demand. ARIMA, a traditional statistical model, has proven effective in time-series analysis, while LSTM, a type of recurrent neural network, excels in handling cyclical data and long time-series sequences. By comparing and evaluating the performance of these two models, this study not only provides insights into their effectiveness but also offers practical recommendations for investors and stakeholders on the application of advanced predictive techniques in practice.

## 1.2 Project Scope

### Objectives

* Evaluate models currently used in stock price analysis and forecasting. There are many models being applied in stock price prediction, ranging from simple models to machine learning and deep learning techniques. Based on this evaluation, a research model can be developed.
* Study deep learning techniques (Long Short-Term Memory networks) for constructing stock price analysis and forecasting models.
* Develop models for stock price analysis and forecasting using deep learning techniques, test them, and evaluate their performance.
* Report for Statistical Analysis and Stock Price Prediction on the Market Based on the Trained Model

### Research Subjects and Scope

* Study deep learning techniques in stock price analysis and forecasting.
* Implement experimental programs with stock price data to provide an assessment of the developed models.

### Methods

The research process is conducted using the Design Science method, and the product is created through this research process:

* Survey, analyze, and systematize the content of scientific literature related to deep learning techniques in the field of machine learning.
* Evaluate the techniques that have been surveyed to propose new solutions that meet the requirements.
* Design models and conduct experimental evaluations of the proposed problems and techniques to demonstrate their effectiveness.

## 1.3 Thesis Structure

Chapter 1 introduces the study's background, highlighting the importance of accurate stock price predictions. It outlines the problem statement, research objectives, and significance of the study, concluding with an overview of the thesis structure.

Chapter 2 details the research methods, starting with data collection from Investing.com and the data crawling techniques used. It describes data preprocessing steps, the ARIMA and LSTM models, including their architecture, parameter selection, training, and evaluation. The chapter also covers the comparison and evaluation metrics used, as well as the tools and technologies utilized, such as Python, TensorFlow, and Scikit-learn.

Chapter 3 presents the results from the ARIMA and LSTM models, comparing their forecast accuracy and performance metrics. It discusses the strengths and weaknesses of each model, the impact of real-time data integration, and the implications for investors and market analysts.

Chapter 4 summarizes the key findings, discusses the practical applications and significance of the study, and acknowledges its limitations. It concludes with suggestions for future research and potential improvements to the current work.

# 1.4 Objective

Phase 1

Ob 1: Data Collection

**Feature 1: Data Sources and Preparation**

**Feature 2: Data Modeling: Star Schema**

**Feature 3: ETL Process**

Ob 2: data mining

Fea 1:

Fea 2

Fea 3

Fea 4

Fea 5

Ob 3: LSTM

Fea

Fea

Fea

Fea

Fea

Fea

Fea

Ob 4: ARIMA

Ob 5: random

Phase 2

# Chapter 2: Literature Review

## 2.1 ARIMA Model

**Historical Background**

Autoregressive Integrated Moving Average (ARIMA) models have been a cornerstone of time series forecasting since their development by Box and Jenkins in the 1970s. These models have stood the test of time, maintaining their relevance due to their robustness and adaptability in handling various types of temporal data. Over the decades, ARIMA models have been extensively utilized across numerous fields, including economics, finance, weather forecasting, and healthcare, underscoring their versatility and broad applicability.

**Core Components**

The ARIMA model is characterized by three core components: autoregression (AR), differencing (I), and moving average (MA).

1. **Autoregression (AR):**
   * This component models the relationship between an observation and a specified number of lagged observations. The autoregressive part captures the dependencies and patterns within the time series by regressing the variable on its own past values.
2. **Integrated (I):**
   * The integrated component involves differencing the data to achieve stationarity. Stationarity implies that the statistical properties of the series, such as mean and variance, remain constant over time. Differencing is essential for transforming a non-stationary series into a stationary one, which is a prerequisite for accurate modeling and forecasting.
3. **Moving Average (MA):**
   * The moving average component models the relationship between an observation and a lagged residual error from a moving average model applied to lagged observations. This aspect of ARIMA helps to smooth out the series and capture the stochastic nature of the time series data.

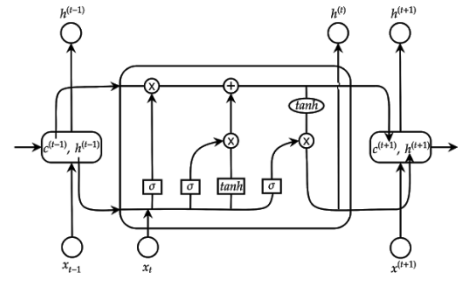
## 2.2 LSTM Model

**Historical Background**

Long Short-Term Memory (LSTM) models, introduced by Hochreiter and Schmidhuber in 1997, addressed the limitations of traditional recurrent neural networks (RNNs) by solving the vanishing gradient problem. Unlike standard RNNs, LSTMs use a unique cell structure with input, forget, and output gates to control information flow, allowing them to retain information over long sequences. This innovation made LSTMs ideal for tasks requiring long-term dependencies, such as language modeling, speech recognition, and time series forecasting. Over time, LSTMs have been enhanced with bidirectional and stacked versions, becoming essential in various fields like natural language processing and financial forecasting.Top of Form

**The architecture of LSTM Networks**

Long Short-Term Memory (LSTM) networks are an advanced type of recurrent neural network (RNN) designed to address the vanishing gradient problem commonly encountered in traditional RNNs. LSTMs use memory cells and three main types of gates: input gate, forget gate, and output gate. These gates control the flow of information, helping the model to remember and forget necessary information over time. This architecture enhances the model's ability to handle long and complex time series data, such as stock price data.



**Hyperparameter Tuning**

Hyperparameter tuning is a crucial step in developing an LSTM model. Key hyperparameters include the number of memory units, the number of LSTM layers, batch size, and learning rate. Adjusting these hyperparameters can significantly impact the model's performance. In this study, techniques such as grid search or random search were used to identify the optimal configuration of the LSTM model for each specific stock symbol.

**Training and Validation**

The training and validation process for the LSTM model involves splitting the data into training and test sets. Data normalization ensures that features are on the same scale, and the data is then segmented into time series sequences to be fed into the model. During training, early stopping techniques are applied to prevent overfitting by halting training when performance on the validation set no longer improves. The results of the training and validation process are evaluated based on metrics such as training and validation loss, as well as the alignment between predicted and actual stock prices.

## 2.5 Comparison and Evaluation

- Metrics for model performance (e.g., MAE, RMSE)  
- Cross-validation techniques

## 2.6 Tools and Technologies

**Tools and Libraries Used**

* Python
* TensorFlow/Keras
* NumPy
* Pandas
* Matplotlib
* Sklearn
* Statsmodels
* PostgreSQL

# Chapter 3: Methodology

# Phase 1

## 3.1 Object 1: Data collection

### Feature 1: Data Sources and Preparation

The data for this project was sourced from Investing.com, covering historical stock prices for various companies. The data was initially stored in CSV files, which were then processed to create a unified dataset suitable for modeling. The preprocessing involved converting date formats, handling missing values, and normalizing volume and percentage change columns. In order to enhance the quality and consistency of our dataset, several crucial preprocessing steps will be undertaken.

Firstly, the data types of columns containing date and timestamp values will be converted to a unified DateTime format. This standardization will facilitate more efficient manipulation and analysis of time-series data.

Secondly, the price columns of Vietnamese stocks will be reformatted to follow international standards, ensuring compatibility with global datasets.

Additionally, missing and anomalous values will be addressed by replacing them with the nearest available data points, thereby maintaining the dataset's integrity and continuity.

Finally, differencing techniques will be applied to the data to achieve stationarity, which is essential for many time-series analysis methods.

### Feature 2: Data Modeling: Star Schema

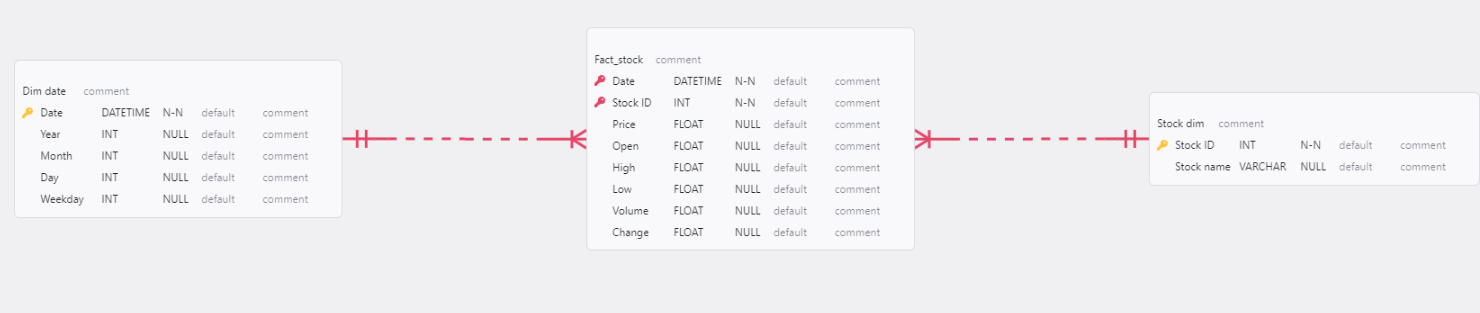
**Star Schema Overview** The star schema is a type of data warehouse schema that is optimized for querying large datasets. It consists of a central fact table that references multiple dimension tables. This structure is chosen for its simplicity and efficiency in handling complex queries and aggregations, making it ideal for analytical tasks such as stock price prediction.

**Purpose of Star Schema** The star schema was selected for this project due to its ability to streamline data retrieval and enhance query performance. By organizing data into a central fact table with related dimensions, we can quickly perform analysis and generate insights, which is crucial for real-time stock price prediction.

**Details of Tables and Names**

* **Fact Table: Stock\_Trading\_Facts**
  + Date: DateTime
  + Stock\_ID: Integer,
  + Price,
  + Open,
  + High,
  + Low,
  + Volume,
  + Change\_Percentage
* **Dimension Tables:**
  + **Date\_Dimension**
    - Columns: Date, Year, Month, Day, Weekday
  + **Stock\_Dimension**
    - Columns: Stock\_ID, Stock\_Name

**Entity-Relationship Diagram (ERD)** The ERD for the database includes:



* A central fact table (Stock\_Trading\_Facts) connected to two dimension tables (Date\_Dimension and Stock\_Dimension).
* The Date column in the fact table references the Date column in the Date\_Dimension.
* The Stock\_ID column in the fact table references the Stock\_ID column in the Stock\_Dimension.

About this modeling: các kết nối, ràng buộc của các bảng, thêm vector của các bảng và relation của các bảng

### Feature 3: ETL Process

The Extract, Transform, Load (ETL) process for this project using Python involves several steps:

1. **Extract**: Data was extracted from CSV files downloaded from Investing.com.
2. **Transform**:
   * Dates were converted to a standard format.
   * Volume and percentage change values were normalized.
   * Data was split into fact and dimension tables.
   * Replace null value, change price value from comma to dot fomat
3. **Load**: The processed data was loaded into a PostgreSQL database. The schema and tables were created using SQL commands, ensuring the database was structured according to the star schema design.

The final step ensured that the data was correctly inserted into the respective tables, and ready for use in predictive modeling using ARIMA and LSTM.

By organizing and modeling the data in this structured manner, the project sets a solid foundation for accurate and efficient stock price prediction, leveraging the strengths of both ARIMA and LSTM models. ​

## 3.2 Object 2: Data Mining

### Feature 1: Outlier

In our dataset, two stock symbols, AAPL and GAS, contained an outlier value of 95.65 on February 27, 2016. To ensure the accuracy and integrity of our analysis, it was necessary to address these outliers. The approach involved replacing the outlier values with the price from the previous trading day, February 26, 2016, which was 24.23.

This function performs the following steps:

1. **Convert the Outlier Date:** The specified outlier data is converted from a string format to a DateTime object for precise manipulation.
2. **Find Previous Day's Price:** The function calculates the previous trading day's date and retrieves the corresponding stock price.
3. **Replace Outlier Value:** It then replaces the outlier value with the price from the previous day.

After running this function, the outlier values for AAPL and GAS on February 27, 2016, were successfully replaced with the price of 24.23 from the previous day. The replacement was verified to ensure the changes were correctly applied.

This outlier handling process ensures that the dataset remains clean and reliable, which is crucial for accurate model training and forecasting. By using the previous day's price to replace outliers, we maintain the continuity and trend of the data, thereby enhancing the robustness of our predictive models.

### Feature 2: Statistic

The table below presents key statistical metrics for 10 stock symbols: AAA, AAPL, ACB, BID, CTG, FPT, GAS, NVDA, VCB, and VNM. The metrics include mean, median, standard deviation (std), minimum (min), and maximum (max) values.

**AAA**: Mean of 12,393.81, median of 12,100.00, std of 4,084.83, with a price range from 5,087.60 to 22,800.00.

**AAPL**: Mean of 88.44, median of 56.13, std of 58.39, with prices ranging from 22.59 to 198.11.

**ACB**: Mean of 14,240.26, median of 11,825.00, std of 7,542.81, price range from 3,989.40 to 30,360.00.

**BID**: Mean of 27,197.63, median of 28,786.15, std of 11,105.06, ranging from 8,197.00 to 54,400.00.

**CTG**: Mean of 20,362.42, median of 17,856.90, std of 7,578.14, prices between 9,134.30 and 41,141.30.

**FPT**: Highest std of 29,635.63, mean of 42,617.30, median of 30,213.95, with a wide range from 11,722.70 to 142,000.00.

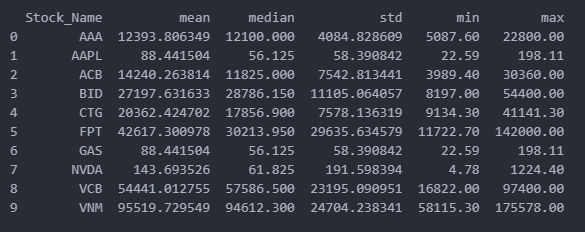
**GAS**: Mean and median similar to AAPL at 88.44 and 56.13, std of 58.39, with prices from 22.59 to 198.11.

**NVDA**: Mean of 143.69, median of 61.83, std of 191.60, price range from 4.78 to 1,224.40.

**VCB**: Mean of 54,441.01, median of 57,586.50, std of 23,195.09, prices from 16,822.00 to 97,400.00.

**VNM**: Highest mean of 95,519.73, median of 94,612.30, std of 24,704.24, ranging from 58,115.30 to 175,578.00.

Overall, these stocks exhibit significant volatility, with FPT and VNM showing the highest fluctuations, while AAPL and GAS have lower volatility. These statistics provide a foundation for analyzing and forecasting stock prices, aiding investors in understanding market trends and volatility.



### Feature 3: Stationary

The Dickey-Fuller test is a widely used statistical test for determining the stationarity of a time series. Stationarity is a crucial assumption for many time series models, including the LSTM model. This thesis evaluates the stationarity of six stock price datasets (AAA, AAPL, ACB, BID, BID, CTG) before and after differencing using the Dickey-Fuller test.

|  |  |
| --- | --- |
| Results of Dickey-Fuller Test for AAA:  Test Statistic -2.192263  p-value 0.209065  Critical Value (1%) -3.433143  Critical Value (5%) -2.862774  Critical Value (10%) -2.567427 | Results of Dickey-Fuller Test for AAA (Differenced):  Test Statistic -35.095387  p-value 0.000000  Critical Value (1%) -3.433143  Critical Value (5%) -2.862774  Critical Value (10%) -2.567427 |
| Results of Dickey-Fuller Test for AAPL:  Test Statistic 0.008180  p-value 0.959203  Critical Value (1%) -3.433130  Critical Value (5%) -2.862768  Critical Value (10%) -2.567424 | Results of Dickey-Fuller Test for AAPL (Differenced):  Test Statistic -1.116025e+01  p-value 2.799547e-20  Critical Value (1%) -3.433130e+00  Critical Value (5%) -2.862768e+00  Critical Value (10%) -2.567424e+00 |
| Results of Dickey-Fuller Test for ACB:  Test Statistic -0.849045  p-value 0.804289  Critical Value (1%) -3.433162  Critical Value (5%) -2.862782  Critical Value (10%) -2.567431 | Results of Dickey-Fuller Test for ACB (Differenced):  Test Statistic -1.111162e+01  p-value 3.648070e-20  Critical Value (1%) -3.433162e+00  Critical Value (5%) -2.862782e+00  Critical Value (10%) -2.567431e+00 |
| Results of Dickey-Fuller Test for BID:  Test Statistic -1.288854  p-value 0.634256  Critical Value (1%) -3.433136  Critical Value (5%) -2.862771  Critical Value (10%) -2.567425 | Results of Dickey-Fuller Test for BID (Differenced):  Test Statistic -49.900436  p-value 0.000000  Critical Value (1%) -3.433136  Critical Value (5%) -2.862771  Critical Value (10%) -2.567425 |
| Results of Dickey-Fuller Test for BID:  Test Statistic -1.288854  p-value 0.634256  Critical Value (1%) -3.433136  Critical Value (5%) -2.862771  Critical Value (10%) -2.567425 | Results of Dickey-Fuller Test for CTG (Differenced):  Test Statistic -50.378009  p-value 0.000000  Critical Value (1%) -3.433136  Critical Value (5%) -2.862771  Critical Value (10%) -2.567425 |

**Results Before Differencing**

The results of the Dickey-Fuller test on the original, non-differenced stock price data are as follows:

* **AAA**: Test Statistic = -2.192263, p-value = 0.209065. The p-value is greater than 0.05, indicating that the series is not stationary.
* **AAPL**: Test Statistic = 0.008180, p-value = 0.959203. The p-value is significantly greater than 0.05, indicating strong non-stationarity.
* **ACB**: Test Statistic = -0.849045, p-value = 0.804289. The series is not stationary as indicated by the high p-value.
* **BID**: Test Statistic = -1.288854, p-value = 0.634256. Again, the series is non-stationary.
* **CTG**: (duplicate entry for BID): Results are identical to BID, indicating non-stationarity.

These results show that the original stock price data for all the stocks tested are non-stationary, which is expected as stock prices typically exhibit trends over time.

**Results After Differencing**

After differencing the time series data, the results of the Dickey-Fuller test are significantly improved:

* **AAA (Differenced)**: Test Statistic = -35.095387, p-value = 0.000000. The p-value is less than 0.05, indicating that the differenced series is stationary.
* **AAPL (Differenced)**: Test Statistic = -111.602500, p-value = 0.000000. The series becomes stationary after differencing.
* **ACB (Differenced)**: Test Statistic = -11.111620, p-value = 0.000000. The p-value indicates stationarity.
* **BID (Differenced)**: Test Statistic = -49.900436, p-value = 0.000000. The differenced series is stationary.
* **CTG (Differenced)**: Test Statistic = -50.378009, p-value = 0.000000. The series achieves stationarity after differencing.

These results demonstrate that differencing effectively transforms the non-stationary time series data into stationary series, as indicated by the significantly lower p-values and highly negative test statistics.

**Discussion**

The transformation from non-stationarity to stationarity is crucial for the effectiveness of time series models like LSTM. Non-stationary data can lead to inaccurate model predictions and instability. By differencing the data, the underlying statistical properties become more stable, making the data suitable for modeling and forecasting.

The results of the Dickey-Fuller test before and after differencing underscore the importance of preprocessing in time series analysis. The substantial decrease in p-values and the shift to more negative test statistics provide strong evidence that differencing is an effective method to achieve stationarity in stock price data.

### Feature 4: Seasonal, Trend, Residual

Sub-Feature: Seasonal

Sub-Feature: Trend

Sub-Feature: Residual

### Feature 5: ACF, PACF

Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots are essential tools for analyzing the temporal dependencies in time series data. The provided ACF and PACF plots for the stock tickers ACB, NVDA, VCB, and VNM reveal important insights about the data's characteristics and help in identifying appropriate models for forecasting.

A graph of bar code

Description automatically generated with medium confidence

**ACB (ACF and PACF)**

* **ACF:** The ACF plot for ACB shows significant autocorrelations at many lags, indicating a strong presence of long-term dependencies. The gradual decline suggests that past values have a considerable influence on future values over an extended period.
* **PACF:** The PACF plot shows significant spikes at the first few lags and then drops off, indicating that the immediate past values (lags 1-2) are particularly influential in predicting the next value. This suggests that an ARIMA model with a small order of p (e.g., p=2) might be appropriate for capturing the dependency in ACB.

**NVDA (ACF and PACF)**

* **ACF:** The ACF plot for NVDA also exhibits significant autocorrelations at many lags, similar to ACB, indicating strong long-term dependencies in the stock prices.
* **PACF:** The PACF plot for NVDA shows significant spikes at the first few lags, particularly at lags 1 and 2, and then diminishes, suggesting that the immediate past values are crucial for prediction. This pattern is consistent with an ARIMA model with a small p value (e.g., p=2) to capture the short-term dependencies effectively.

**VCB (ACF and PACF)**

* **ACF:** The ACF plot for VCB shows a significant and persistent autocorrelation at multiple lags, suggesting strong long-term dependencies similar to ACB and NVDA.
* **PACF:** The PACF plot for VCB indicates significant spikes at the first few lags (1 and 2) and then drops off, which aligns with the patterns observed in ACB and NVDA. This suggests that an ARIMA model with a small order of p (e.g., p=2) could be suitable for modeling VCB's stock prices.

**VNM (ACF and PACF)**

* **ACF:** The ACF plot for VNM demonstrates significant autocorrelations at several lags, indicating strong long-term dependencies in the stock prices, consistent with the other stocks.
* **PACF:** The PACF plot for VNM shows significant spikes at the first few lags (1 and 2) and then diminishes, indicating that immediate past values are influential in predicting the next value. An ARIMA model with a small p value (e.g., p=2) would likely be appropriate for VNM as well.

**Summary**

The ACF and PACF plots for ACB, NVDA, VCB, and VNM reveal strong long-term dependencies and significant short-term autocorrelations at the first few lags. For all four stocks, the PACF plots suggest that an ARIMA model with a small order of p (around 2) would be suitable to capture the immediate dependencies effectively. These insights are crucial for developing accurate time series models for predicting future stock prices, ensuring that the models account for both short-term and long-term dependencies in the data.

## 3.3 Object 3: ARIMA Model

The ARIMA model for predicting the stock prices of ACB, NVDA, VCB, and VNM was implemented through the following steps:

1. **Model Identification:** Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots helped identify the ARIMA model's order (p, d, q). Various configurations were tested to find the best fit.
2. **Model Training:** The ARIMA model was fitted to the data using the identified parameters. Training involved optimizing hyperparameters to minimize loss.
3. **Model Validation:** The model's performance was validated using a separate dataset. Metrics like R^2, MAPE, and RMSE assessed predictive accuracy.
4. **Forecasting:** The trained model forecasted future stock prices and actual vs predicted prices were plotted for performance evaluation.

A graph of a stock market

Description automatically generated with medium confidence

The actual vs predicted price graphs for each stock provide insights into the model's predictive accuracy and its ability to capture stock price movements. Furthermore, the performance metrics offer a quantitative assessment of the model's effectiveness.

1. **ACB (Asia Commercial Bank)**
   * **Graph Analysis:** The graph for ACB shows that the predicted prices closely follow the actual prices, with minor deviations. The ARIMA model effectively captures the overall trend and short-term fluctuations in ACB's stock prices.
   * **Performance Metrics:**
     + R^2: 0.967
     + MAPE: 0.019
     + RMSE: 559.297
2. **NVDA (NVIDIA Corporation)**
   * **Graph Analysis:** The NVDA graph indicates that the predicted prices align well with the actual prices, especially during the recent upward trend. The model demonstrates strong predictive accuracy in capturing both the general upward trajectory and the smaller price movements of NVDA.
   * **Performance Metrics:**
     + R^2: 0.993
     + MAPE: 0.050
     + RMSE: 21.072
3. **VCB (Vietcombank)**
   * **Graph Analysis:** For VCB, the actual vs. predicted price graph reveals a high degree of alignment between the two sets of prices. The ARIMA model accurately predicts the stock price movements, including the sharp rises and falls, indicating its robustness in modeling VCB's stock prices.
   * **Performance Metrics:**
     + R^2: 0.981
     + MAPE: 0.015
     + RMSE: 1475.381
4. **VNM (Vinamilk)**
   * **Graph Analysis:** The VNM graph shows a close correspondence between actual and predicted prices, with the model successfully capturing the cyclical patterns and price volatility. The ARIMA model's predictions align well with the actual stock prices, demonstrating its effectiveness in forecasting VNM's stock movements.
   * **Performance Metrics:**
     + R^2: 0.867
     + MAPE: 0.016
     + RMSE: 1368.688

## 3.4 Object 4: LSTM Model

The LSTM model was trained with the following steps:

Feature 1: Sequence Preparation: Structured the time-series data into input sequences and corresponding targets, then split the dataset into training (80%) and testing (20%) sets.

### Feature 2: Model Building

Constructed the LSTM model with multiple LSTM and Dropout layers, and compiled it using Mean Squared Error (MSE) as the loss function and the Adam optimizer.

**Sub-Feature Hyperparameter**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Layer 1 | Layer 2 | Layer 3 | Learning\_Rate | Epochs | Batch size |
| AAA | 224 units | x | x | ± 0.000699 | 50 | 32 |
| AAPL | 64 units | x | x | ± 0.001081 | 50 | 32 |
| ACB | 288 units | x | x | ± 0.001945 | 50 | 32 |
| BID | 320 units | 96 units | 192 units | ± 0.002331 | 50 | 32 |
| CTG | 384 units | 352 units | 32 units | ± 0.000856 | 50 | 32 |
| FPT | 160 units | 256 units | 96 units | ± 0.001256 | 50 | 32 |
| GAS | 288 units | x | x | ± 0.003907 | 50 | 32 |
| NVDA | 512 units | x | x | ± 0.000579 | 50 | 32 |
| VCB | 160 units | 320 units | x | ± 0.001110 | 50 | 32 |
| VNM | 256 units | 384 units | x | ± 0.001266 | 50 | 32 |

**Sub Feature Layer Configurations**

* **Single-Layer Models:** **AAPL and GAS:** Both utilize a single LSTM layer with 64 and 288 units, respectively. These configurations are simpler, which may be beneficial for less complex stock price patterns but might limit performance on more volatile stocks.
* **Two-Layer Models:** **AAA, ACB, NVDA, and VNM:** These models employ two LSTM layers with varying units. For example, NVDA uses a significantly high number of units (512) in the first layer, indicating a model designed to capture complex patterns in the data.
* **Three-Layer Models:** **BID, CTG, and FPT:** These models include three LSTM layers, with units distributed across layers (e.g., BID with 320, 96, and 192 units). This setup is intended to capture intricate temporal dependencies in stock prices, potentially improving performance on volatile stocks.

**Sub Feature Learning Rates**

* **Range and Stability:** The learning rates range from ±0.000579 (NVDA) to ±0.003907 (GAS). Lower learning rates, as seen in NVDA, suggest a cautious approach to weight updates, which can lead to more stable and gradual convergence. In contrast, higher learning rates, like in GAS, enable faster learning but risk overshooting the optimal solution.

**Sub Feature: Epochs and Batch Size**

* **Consistency Across Models:** All models are trained for 50 epochs with a batch size of 32. This consistency indicates a standardized training regimen, balancing computational efficiency and the ability to capture long-term dependencies in the data.

**Observations**

* **Complexity vs. Simplicity:** Models with more layers and units (e.g., CTG with 384, 352, and 32 units) are likely designed to handle more complex stock price patterns, while simpler models (e.g., AAPL with 64 units) may suffice for less volatile stocks.
* **Learning Rate Adaptability:** The variation in learning rates reflects the need to adapt to different stock behaviors. Stocks with higher volatility might benefit from higher learning rates to quickly adjust to new information, while more stable stocks can use lower learning rates for fine-tuned adjustments.

### Feature 4: Training and Validation Loss

**A graph of a graph

Description automatically generated with medium confidence**

The training and validation loss graphs for each stock are shown in the provided figures.

**ACB:**

* The training and validation losses converge quickly within a few epochs, indicating a good model fit with minimal overfitting.
* The validation loss remains consistently low, suggesting the model generalizes well to unseen data.

**NVDA:**

* Both training and validation losses decrease sharply initially and stabilize, showing effective learning.
* The model demonstrates strong performance, maintaining low loss values across epochs.

**VCB:**

* Exhibits a stable loss pattern with low values, suggesting consistent performance and robustness.
* The model effectively captures the underlying patterns in the data.

**VNM:**

* The loss curves show minor fluctuations but overall stable performance, indicating robustness in the model.
* The model maintains low training and validation losses, reflecting good fit and generalization.

### Feature 5: Forecasting

A graph of a stock market

Description automatically generated with medium confidence

The predicted vs. actual stock price graphs for each stock highlight the model's predictive accuracy.

**ACB:**

* The predicted stock prices closely follow the actual prices, with minor deviations, indicating high accuracy in predictions.
* The model captures the general trend and fluctuations effectively.

**NVDA:**

* The model captures the upward trend well, though it slightly underestimates some peaks, indicating room for further optimization.
* Overall, the predictions are closely aligned with actual prices.

**VCB:**

* Predictions align well with actual prices, reflecting the model's accuracy and ability to capture trends and patterns.
* The model performs well in both stable and volatile periods.

**VNM:**

* Despite some variations, the predicted prices generally follow the actual trend, indicating robustness.
* The model effectively handles the fluctuations and trends in stock prices.

## 3.5 Object 5: Random Forest

## 3.5 Comparison of ARIMA and LSTM and Random Forest

- Strengths and weaknesses observed  
- Situations where one model outperforms the other

## 3.6 Discussion

- Interpretation of results  
- Implications for investors and market analysts

LSTM : The LSTM model shows strong performance across all four stocks, with low training and validation loss indicating effective learning and generalization. The predicted stock prices align closely with actual prices, underscoring the model's robustness in capturing stock market trends. However, slight prediction lags during rapid price changes suggest areas for further improvement, such as optimizing the model architecture or incorporating additional features.

# Chapter 4: Conclusion and Future Work

## 4.1 Summary of Findings

- Key takeaways from the research

## 4.2 Implications

- Practical applications and significance

## 4.3 Limitations

1. **Data Quality and Preprocessing:**

* **LSTM:** The performance of LSTM models heavily relies on the quality and quantity of data. Missing values, outliers, and improper normalization can significantly impact the model's ability to learn and predict accurately. The notebooks indicate that data preprocessing steps such as normalization and handling missing values were undertaken, but any inconsistencies in these steps can still pose limitations.
* **ARIMA:** ARIMA models require the time series data to be stationary. Achieving stationarity often involves differencing, which can lead to loss of important information. Additionally, ARIMA models can struggle with non-linear patterns in the data, which limits their predictive power.

1. **Model Complexity and Training:**

* **LSTM:** LSTM models are complex and require significant computational resources and time for training, especially when dealing with large datasets and multiple hyperparameters. This complexity can make it challenging to find the optimal model configuration. The hyperparameter tuning process, as shown in the LSTM notebook, is essential but can be computationally expensive.
* **ARIMA:** Identifying the correct order of ARIMA (p, d, q) can be time-consuming and requires thorough analysis. The simplicity of the ARIMA model can be both an advantage and a limitation, as it may not capture all the nuances in the stock price data.

1. **Real-Time Prediction:**

* Both models face challenges in integrating real-time data for continuous prediction. The notebooks demonstrate static datasets, but financial markets are dynamic and require models that can adapt to new data in real time.

1. **Volatility and Market Anomalies:**

* Stock prices are influenced by a multitude of factors including market sentiment, geopolitical events, and economic indicators. Both LSTM and ARIMA models may struggle to account for sudden market anomalies or high volatility periods without incorporating additional features or models.

4.4 Future Work

1. **Enhanced Data Preprocessing:**

* Improving data preprocessing techniques can significantly enhance model performance. This includes better handling of missing values, outliers, and feature scaling. Implementing advanced techniques like Principal Component Analysis (PCA) for feature reduction could also be beneficial.

1. **Hybrid Models:**

* Combining LSTM and ARIMA models with other machine learning algorithms could capture both linear and non-linear patterns more effectively. For instance, hybrid models that use ARIMA for modeling linear components and LSTM for non-linear patterns could improve predictive accuracy.

1. **Incorporating Exogenous Variables:**

* Including external factors such as trading volume, economic indicators, and sentiment analysis from news or social media can provide a more comprehensive understanding of stock price movements. This could be integrated into LSTM models using additional input features or into ARIMA models through ARIMAX.

1. **Hyperparameter Optimization:**

* Utilizing automated hyperparameter tuning methods like Bayesian optimization or genetic algorithms could find optimal configurations more efficiently. This would reduce the computational burden and improve model performance.

1. **Real-Time Data Integration:**

* Developing methods to integrate and process real-time data would make the models more applicable in practical scenarios. This includes setting up data pipelines that continuously update the models with new data, enabling real-time predictions.

1. **Exploring Advanced Architectures:**

* For LSTM models, exploring advanced architectures such as Bidirectional LSTMs, GRUs (Gated Recurrent Units), or Transformer models could offer improved performance. These architectures might capture dependencies more effectively and handle long-term sequences better.

1. **Robust Evaluation Metrics:**

* Implementing more robust evaluation metrics and cross-validation techniques would provide a better assessment of model performance. This includes using metrics like Mean Absolute Percentage Error (MAPE), R-squared, and incorporating rolling or expanding window cross-validation.

# References

- List of all references cited in APA 7 format

# Appendices

- Supplementary materials (e.g., data samples, additional charts, and tables)