**Stock Price Gap Prediction with Deep Learning**

**PRANAY ULHAS**

**Student ID: 2293112**

Supervised by Dr. Vitaliy Milke

A logo with a bird and a crown

Description automatically generated

A THESIS SUBMITTED IN FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE MSc ARTIFICIAL INTELLIGENCE

SCHOOL OF COMPUTING AND INFORMATION SCIENCES

ANGLIA RUSKIN UNIVERSITY

13/9/2024

**Acknowledgment**

I would like to express my deepest gratitude to my supervisor, Dr. Vitaliy Milke, for his unwavering support, guidance, and encouragement throughout the course of this project. His insightful feedback and expertise were invaluable in shaping the direction of my research and ensuring its successful completion. I am truly grateful for the time and effort he invested in helping me navigate through complex challenges and expand my knowledge in the field of deep learning and financial forecasting.

I would also like to extend my appreciation to my family and friends, whose continuous encouragement and patience motivated me to keep pushing forward during the more difficult moments of this journey. Finally, I would like to thank the institution and all the faculty members who provided me with the necessary tools and resources to complete this project.

Thank you all for your unwavering support and belief in my abilities.

**TABLE CONTENTS:** page no

1. Abstract ……………………………………………………………………………….6

2. Introduction…………………………………………………………………………..8

2.1 Background and Motivation…………………………………………………8

2.2 Problem Statement…………………………………………………………….9

2.3 Objectives…………………………………………………………………………10

2.4 Structure of the Report………………………………………………………..11

3. Aim……………………………………………………………………………………….12

4. Literature Review…………………………………………………………………….12

4.1 Overview of Price Gaps in Financial Markets………………………….12

4.2 Types of Price Gaps……………………………………………………………14

4.2.1. Common Gaps……………………………………………………………14

4.2.2. Breakaway Gaps…………………………………………………………14

4.2.3. Runaway (Continuation) Gaps………………………………………15

4.2.4. Exhaustion Gaps………………………………………………………..15

4.3 Historical Perspective on Price Gap Studies………………………….15

4.4. Traditional Methods: Predictive Modelling in Finance…………….17

4.4.1. Moving Averages…………………………………………………………18

4.4.2. Autoregressive Models…………………………………………………18

4.4.3. Inability to Handle High Volatility……………………………………19

4.4.4. Overfitting to Historical Data…………………………………………20

4.5. Limited Incorporation of Market Sentiment and Behavioural Factors………………………………………………………………………………….20

4.6. Inspiration from Machine Learning Techniques………………………21

5. Methodology………………………………………………………………………….22

5.1. Data Preprocessing……………………………………………………………22

5.2. Model Selection and Architecture………………………………………..23

5.3. Training and Testing……………………………………………………………24

5.4. Model Evaluation……………………………………………………………….26

5.5. Tools and Technologies……………………………………………………….26

5.6. Limitations………………………………………………………………………..29

6.Design and Development………………………………………………………….31

6.1. System Design…………………………………………………………………..31

6.2. Development Process…………………………………………………………32

6.2.1.Data Preprocessing………………………………………………………….33

6.2.2. Train-Test Split…………………………………………………………………33

6.2.3.Model Definition (LSTM/GRU)……………………………………………..33

6.2.4. Model Training…………………………………………………………………34

6.2.5. Model Evaluation……………………………………………………………..36

6.2.6. Visualization……………………………………………………………………36

6.3. Data Gathering Approaches…………………………………………………36

6.4. Key Aspects of the System…………………………………………………..37

6.5. Algorithms Underpinning the Design……………………………………..37

6.6. Rejected Algorithms……………………………………………………………37

6.7. Summary of the Design and Development Process…………………38

7. Implementation and Testing……………………………………………………..40

7.1. Translating Design and Pseudo-Code into Actual Code……………41

7.2. Key Parts of Code with Explanations……………………………………..41

7.3 Testing and Validation Approach…………………………………………..42

7.3.1. Validation Approach………………………………………………………..43

7.3.2. Testing Approach…………………………………………………………….43

7.3.3. Evaluation and Results……………………………………………………..43

7.3.4. Partial and Full Compilation Results……………………………………47

7.3.5. Visualization of Results…………………………………………………….47

7.4. Findings and Results …………………………………………………………..48

7.4.1. Accuracy over Epochs………………………………………………………49

7.4.2. Confusion Matrix and Classification Metrics………………………..49

7.4.3. Actual vs Predicted Gap Closures……………………………………….51

7.4.4. Conclusion of Findings……………………………………………………..51

8. Discussion and Critical Appraisal of Results………………………………..51

8.1. Interpretation of Findings in the Context of the Literature Review..52

8.2. Usefulness and Relevance of the Findings………………………………56

8.3. Statistical Significance of the Results…………………………………….57

8.4. Significant Aspects of the Findings………………………………………..58

8.5. Critical Appraisal and Areas for Improvement………………………….59

9. Conclusions and Further Work…………………………………………………..60

9.1. Summary of Project Outcomes……………………………………………..61

9.2. Key Technical Results and Conclusions………………………………….61

9.3. Meeting the Original Requirements………………………………………..62

9.4. Reflection on Learning and Skills Gained………………………………..63

9.5. Future Work and Development………………………………………………64

10. References…………………………………………………………………………….66

11. Appendix……………………………………………………………………………….68

**1.Abstract**

This project aims to predict stock price gaps for Apple Inc. (AAPL) using advanced deep learning techniques, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) neural networks. Traders and investors are particularly interested in stock price gaps, which are differences between a stock's closing price on one trading day and its opening price on the next. These pauses are frequently caused by events that take place after business hours, such as macroeconomic reports, corporate earnings announcements, and unanticipated geopolitical events, which can present risks or opportunities for profit. Due to the intricate, erratic, and non-linear nature of financial markets—a setting in which conventional statistical techniques frequently falter—predicting such gaps is difficult.

The purpose of this project is to investigate the potential of LSTM and GRU models in predicting stock price gaps for AAPL, using the AAPL/USD exchange rate as a central feature in the model. The AAPL/USD exchange rate was chosen because of its impact on AAPL stock prices, which reflect not only the company's performance but also broader market trends. The dataset used for this project, obtained from Dukascopy Bank's JForex platform, contains historical time series data for AAPL/USD, including high-frequency financial information critical for capturing short-term trends and market shifts.

A comprehensive methodological framework is used, beginning with data preprocessing to clean and transform the raw dataset into a format appropriate for time-series analysis. Missing value handling, data scaling, feature extraction, and lagged variable construction are all important preprocessing steps that help models capture temporal dependencies. The models are then trained to predict the opening price and the size of the price gap using historical price data and other derived features. Hyperparameter tuning is used to optimise the models' architectures, with the goal of achieving the best possible prediction performance.

The LSTM and GRU models were chosen for their ability to handle sequential data and capture long-term dependencies, which are essential for understanding the complex dynamics of financial time series. These models are well-suited to dealing with the irregularities and non-linear patterns that are frequently observed in stock market data. Various metrics were used during the training process to evaluate the models' performance in predicting stock price gaps, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE).

The experimental results show that both LSTM and GRU models can predict stock price gaps with reasonable accuracy, with the GRU model consistently outperforming the LSTM model by a small margin. The GRU model was more efficient at detecting price swings in the short term and exhibited more stable shift during the training process. On the other hand, the LSTM model showed higher accuracy when predicting long-term trends, but struggled to capture rapid shifts in prices as effectively as the GRU.

The results show that the two models are very sensitive to the input feature selection and hyperparameter configuration. Key variables such as historical price series, volume, and technical indicators all played important roles in improving model accuracy. The inclusion of lagged values and technical indicators improved the models' ability to recognise recurring patterns in the data, which contributed to their predictive accuracy.

The study's results demonstrate the promise of deep learning models for stock price forecasting, especially for gap prediction, an area where conventional linear models frequently fall short. Accurate predictions require the deep learning models to be able to identify long-term dependencies and non-linear relationships in the financial data. Furthermore, this study identifies directions for future investigation, such as the investigation of new features like sentiment analysis in news, which could enhance the models' predictive power.

Proficiency in deep learning model development, especially in the context of time series forecasting, is one of the skills acquired through this project. More advanced skills in data preprocessing, feature engineering, and hyperparameter optimisation were developed, as well as proficiency with Python-based libraries such as TensorFlow and Keras. This project also shed light on the practical difficulties of implementing machine learning in the finance industry, such as handling irregularities and volatile data. It also offered views on financial time series analysis.

To sum up, the project shows that both LSTM and GRU models can be used to predict stock price gaps, with GRU slightly outperforming LSTM in terms of computational efficiency and prediction accuracy. However, by adding more intricate features, like sentiment analysis or macroeconomic indicators, prediction accuracy can be further increased. For traders and financial analysts looking to incorporate deep learning models into their predicting tools for making well-informed trading decisions, these findings have real-world applications. To optimise model architectures and expand this work to other stock investments and financial markets, more research is encouraged.

**2. Introduction**

**2.1 Background and Motivation**

Research on financial markets has always been extensive, particularly in the area of stock price prediction, which presents a significant difficulty for traders, investors, and investment managers. Precise prediction of stock prices can assist in making decisions, reduce risks, and pinpoint lucrative trading opportunities. Stock price gaps, or instances where a stock's closing price on one day differs significantly from its opening price on the next, are among the various phenomena that occur in financial markets. These gaps are significant because they can indicate abrupt shifts in market sentiment or outside events.

The reasons behind stock price gaps are frequently corporate announcements, after-hours trading, macroeconomic data releases, and geopolitical events that affect stock prices prior to market opening. Although these gaps can shed light on market behaviour, they also make forecasting more difficult. Forecasting these gaps accurately can help traders take advantage of quick changes in the market or modify their plans to prevent losses. However, gap prediction is a challenging task for traditional forecasting of time series methods, which frequently rely on linear assumptions, due to the intrinsic complexity and volatility of stock markets.

Machine learning and deep learning models are emerging as exciting tools for financial forecasting due to the increasing accessibility of high-frequency data and the rapid development of powerful computation methods. These models have demonstrated success in capturing long-term dependencies and non-linear relationships within time series data, especially Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). Because stock price gaps are influenced by a multitude of interrelated factors, they are therefore well-suited for prediction.

Because of its strong relationship to the price of Apple Inc.'s stock, the AAPL/USD exchange rate is chosen as a major predictor in this project. Being one of the most traded stocks on a global scale, AAPL's movements frequently mirror larger market trends. By concentrating on this financial tool, the project hopes to create a predictive model that can precisely predict price gaps and provide information that individual traders and institutional investors may find helpful.

**2.2 Problem Statement**

Because stock price movements are multifactorial and market behaviour is complex, predicting gaps in stock prices is an important task in financial forecasting. Price gaps are frequently caused by unforeseen outside events that conventional statistical frameworks find difficult to explain, such as economic reports, earnings announcements, or shifts in geopolitics. Furthermore, gaps frequently coincide with increased volatility, which further complicates accurate prediction.

Time series forecasting has long been done using conventional techniques like GARCH (Generalised Autoregressive Conditional Heteroskedasticity) and ARIMA (AutoRegressive Integrated Moving Average), but these approaches frequently fall short of capturing the non-linear dynamics found in stock market data. These models are less useful in complicated and unstable market environments because they make the assumption that past prices will have a greater influence on future price movements.

On the other hand, deep learning models provide a more adaptable method for financial time series analysis, especially those that are built on Recurrent Neural Networks (RNNs), like GRU and LSTM. These models are more adept at predicting stock price gaps because they can extract complex patterns from enormous volumes of data, like short-term fluctuations and long-term dependencies. The creation of a deep learning model that can precisely forecast stock price gaps for AAPL using the AAPL/USD exchange rates as a predictor is the main issue this project attempts to solve.

**2.3 Objectives**

This project's main goal is to create and apply deep learning techniques that can predict Apple Inc.'s stock price gaps with accuracy (AAPL). The project's specific goals are to:

1. **Preprocess and examine the AAPL/USD** dataset in order to get it ready for use in time series forecasting. This covers generating suitable input features, scaling values, and handling missing data.
2. **Create and train LSTM and GRU** models to predict stock price gaps by utilising their capacity to identify intricate relationships and long-term dependencies in financial data..
3. **Compare the efficacy of the LSTM and GRU** models by assessing model performance using popular regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)..
4. **Examine the models' predictive power,** paying particular attention to how well they can forecast price gaps' size and direction as well as how resilient they are to various market circumstances.
5. **Gain understanding of the variables influencing stock price gaps** by looking at the ways in which various market events affect gap occurrences and the ways in which deep learning models can account for these influences.

**2.4 Structure of the Report**

This report is structured as follows:

* **Section 2**: Literature Review examines previous research on stock price forecasting, focusing on traditional time series models as well as modern machine learning and deep learning techniques. It also explores the challenges and potential of predicting stock price gaps.
* **Section 3**: Methodology details the deep learning models implemented, the data preprocessing steps taken, and the specific architecture of the LSTM and GRU networks. It explains the rationale for choosing these models and the process of training and evaluating them.
* **Section 4**: Data Exploration provides an in-depth analysis of the AAPL/USD dataset, highlighting key features, trends, and challenges encountered during data preprocessing. This section also discusses any patterns identified through exploratory data analysis (EDA).
* **Section 5**: Model Development and Results presents the process of training the LSTM and GRU models, along with the results of their performance in predicting stock price gaps. It includes a comparison of the models and a discussion of their strengths and weaknesses.
* **Section 6**: Discussion interprets the findings of the model evaluation, exploring how well the models performed under different market conditions and identifying potential areas for improvement. It also discusses the practical implications of the findings for traders and analysts.
* **Section 7**: Conclusion summarizes the main findings of the project, drawing conclusions about the effectiveness of LSTM and GRU models for stock price gap prediction. This section also outlines potential directions for future research, including the integration of additional features and the application of the models to other financial instruments.

**2.5. Ethics Statement**

I confirm that I have completed the required Ethics training as part of this research project. Additionally, the necessary Ethics form has been submitted, and approval for the project has been granted under the Ethics approval code **ETH2324-9688**. This approval ensures that the research adheres to the ethical standards and guidelines set by the institution, and all procedures involving data collection and analysis were conducted in accordance with these standards.

**3. Aim**

The main goal of this study is to create a solid deep learning model that, when applied to historical financial data, can reliably forecast when stock price gaps for AAPL/USD will close. Through the use of cutting-edge neural network architectures, including Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM), this project aims to increase the accuracy and dependability of forecasts for notable price gap closures. In order to produce useful insights that traders and financial analysts can greatly benefit from, this study attempts to integrate the best aspects of contemporary machine learning techniques with the historical strengths of financial analysis.

**4.LITERATURE REVIEW**

**4.1 Overview of Price Gaps in Financial Markets**

When a stock's closing price on one trading day differs significantly from its opening price on the next, this is known as a price gap. This pricing disparity may be the consequence of a number of variables that affect investor sentiment outside of regular trading hours, such as earnings reports, after-hours news, or general market trends. When there is a price gap, the dynamics of supply and demand have significantly changed during off-market hours, which causes the stock price to move sharply when the market reopens.

Price differences in the financial markets can be very important indicators for investors and traders. They may indicate possible opportunities or hazards and frequently highlight a stock's underlying volatility. For example, an unexpectedly good news story could cause the stock price to rise sharply, indicating higher demand. On the other hand, bad news—like subpar earnings or unstable geopolitical conditions—may cause a gap downward, signalling a decline in market confidence.

For technical analysts, price gaps are especially important because they can reveal information about the momentum and direction of a stock's movement. Since gaps often "close" over time, indicating that the stock may eventually revert to its initial price level, traders frequently use these gaps to forecast short-term price movements. But not all gaps close, and one of the main problems with gap trading strategies is figuring out which ones won't. Making wise trading decisions requires an understanding of the causes and consequences of these gaps because gaps frequently cause abrupt price movements that can result in either significant gains or losses.

Price gaps are essentially indications of market inefficiencies that traders can take advantage of. Their importance stems from their capacity to offer trading strategy entry/exit points as well as signals of potential future price movement. Therefore, for traders and financial analysts hoping to profit from short-term stock price movements, researching price gaps in financial markets—and more importantly, anticipating their closure—is an essential field of study.

Img: GAPS BETWEEN DIFRRENT OPENING AND CLOSING

**4.2 Types of Price Gaps**

Financial market price gaps fall into a number of different categories, each with its own traits and effects on fluctuations in stock prices. Comprehending these categories aids analysts and traders in deciphering market signals and arriving at better trading choices. There are four main categories of price gaps:

**4.2.1. Common Gaps**

The most common kind of price gap, referred to as a trading gap, is common and usually appears in markets with low volatility or relative stability. These gaps are typically tiny and appear in trading ranges devoid of notable price trends or major events. Common gaps are less significant for long-term trends because they are frequently filled quickly, returning the price to its prior levels.

**Consequences:** Common gaps are typically interpreted as transient anomalies that don't portend a major shift in the market or trend. They are not usually used by traders for long-term trading strategies, though they can be useful in predicting short-term fluctuations.

**4.2.2. Breakaway Gaps**

When a stock price breaks out of a trading range or consolidation zone, it's known as a breakaway gap. This phenomenon is frequently caused by a big news story, an earnings report, or other noteworthy developments. Depending on which way the gap is pointing, these gaps usually indicate the beginning of a new trend, either upward or downward.

**Consequences:** Breakaway gaps are thought to be reliable markers of the emergence of a new trend. A breakaway gap is frequently seen by traders as an indication to enter a trade in that direction. If the gap appears following an extended period of sideways trading, it can indicate that a significant price movement is about to begin.

**4.2.3. Runaway (Continuation) Gaps**

Runaway gaps, sometimes referred to as continuation gaps, show the market's persistent momentum in the direction of the dominant trend and appear during strong price trends. When more players enter the market and feed the trend further, these gaps are frequently the result of increased buying or selling pressure.

**Consequences:** Runaway gaps are generally interpreted as proof positive of an established pattern. Assuming the trend continues, traders might use them to bolster their positions. Since these gaps reflect continuing momentum, they are less likely to be filled quickly.

**4.2.4. Exhaustion Gaps**

When a strong trend is coming to an end, exhaustion gaps appear and indicate that the trend is losing steam. These gaps occur when there's a last-minute rush to buy or sell before the trend turns around. High trading volumes frequently accompany exhaustion gaps, indicating that traders are winding down positions.

**Consequences:** It is believed that exhaustion gaps indicate a possible trend reversal. These gaps are seen by traders as alerts to reduce holdings or get ready for a change in the market's trajectory. The price frequently retraces or reverses after an exhaustion gap, closing the gap.

**4.3 Historical Perspective on Price Gap Studies**

Price gap analysis has a long history in the financial literature, going all the way back to the early 20th-century techniques of technical analysis. Traders and analysts noticed price gaps for the first time in an attempt to comprehend market inefficiencies and how these gaps might indicate changes in stock price movements. The main goal of early research was to record price gaps and their apparent correlation with investor behaviour, market sentiment, and significant news events.

**Early Technical Analysis and Price Gaps**

Charles Dow, the creator of Dow Theory, wrote one of the key books for comprehending price differences. Dow did not coin the term "price gap," but his theory emphasised the significance of market trends and price patterns, which prepared the way for the subsequent investigation of gaps as anomalies in price data. Early technical analysts improved upon his theories by viewing gaps as indicators of impending price movements, particularly with regard to spotting trends and reversals.

The idea of price gaps was further developed in the 1930s and 1940s by technical analysts like Richard W. Schabacker and Robert D. Edwards, who co-wrote Technical Analysis of Stock Trends (1948). Price gaps frequently occur during changes in market psychology, reflecting periods of intense buying or selling, as Schabacker's work on chart patterns discussed. The classification of gaps was first proposed by Edwards and Magee, who divided them into four categories that are still in use today: common, breakaway, runaway, and exhaustion gaps.

**Mid-20th Century and Quantitative Studies**

As computers and statistical techniques became more widely available, studies of price differences began to become more quantitative by the middle of the 20th century. The empirical characteristics of price gaps—such as their frequency, size, and duration under various market conditions—began to be studied by researchers. During this time, Eugene Fama popularised the Efficient Market Hypothesis (EMH), which postulated that price differences were merely the result of new information entering the market. Price gaps were arbitrary and unpredictable, according to EMH, because markets react quickly to news.

Not every researcher, though, shared this viewpoint. Price gaps were investigated as part of market anomalies that suggested inefficiencies in studies like those by Benoit Mandelbrot, who popularised the idea of fractal markets, and later by Burton Malkiel, author of A Random Walk Down Wall Street (1973). These studies investigated whether gaps could be predicted and taken advantage of in specific situations, or if they were completely random.

**Modern Research and Machine Learning Approaches**

Price gap analysis has seen a resurgence in popularity in recent decades due to the development of advanced statistical models, high-frequency trading (HFT), and machine learning techniques. Scholars have investigated the possibility that sophisticated models can forecast the emergence and closure of price gaps, especially when it comes to short-term trading tactics. Empirical research has revealed, for instance, that price gaps resulting from major events—like macroeconomic data releases or earnings announcements—tend to be more persistent, whereas smaller gaps frequently close faster.

Financial theorists like Robert Engle, whose ARCH and GARCH models aid in explaining price volatility following gaps, have conducted important studies on price action and volatility clustering. These studies are important contributions to modern gap analysis. In recent times, the application of machine learning algorithms such as Long Short-Term Memory (LSTM) networks has been utilised to forecast the closure of a price gap by analysing past price data and additional technical indicators. More advanced tools for analysing price gaps in real-time markets are now available to traders thanks to these advancements.

**4.4. Traditional Methods: Predictive Modelling in Finance**

Predicting changes in stock prices has been a major area of interest for traders, analysts, and researchers in the finance industry. Over the years, conventional techniques for predicting stock prices have developed, primarily depending on past price data, mathematical models, and market dynamics. Traditional techniques that are still in high demand include technical analysis, autoregressive models, and moving averages. These methods continue to offer helpful insights into price trends and market behaviour, even though they might not always fully capture the complexities of contemporary financial markets. They are based on statistical and economic theory.

**4.4.1. Moving Averages**

One of the easiest and most popular methods for predicting changes in stock prices is the moving average. By figuring out a stock's average price over a predetermined amount of time, such as 50 or 200 days, a moving average smoothes out price data. Longer-term trends are highlighted and short-term fluctuations are helped to filter out.

* **Simple Moving Average (SMA)**: The SMA, or simple moving average, is the most basic type and is determined by averaging the closing prices over a given period of time. The average closing price over the previous 50 days, for instance, would be the 50-day SMA. Moving averages are used by traders to detect trends. For example, a bullish signal is typically indicated when a short-term SMA crosses above a long-term SMA, or a "golden cross."
* **Exponential Moving Average (EMA):** The EMA is more sensitive to fresh information because it places greater weight on recent prices. This is particularly helpful in markets that move quickly, where price changes as of right now are deemed more significant than historical data.

**Applications**: Buy and sell signals are frequently produced by moving averages. Crucial markers of trend reversals are crossovers, which occur when two moving averages of varying lengths cross. Levels of support and resistance are also identified using moving averages.

**4.4.2. Autoregressive Models**

Statistical tools called autoregressive models are used to forecast stock prices in the future based on historical data. These models have a time-dependent structure and predict that future price movements will be based on past values. The Autoregressive Integrated Moving Average (ARIMA) model and its variants are the most widely utilised models in this category.

* **AR (Autoregressive) Models**: Future stock prices are estimated using an AR model by adding up all of the previous prices. For example, a first-order autoregressive model (AR(1)) predicts the next price based on the price that happened right before it.
* **ARIMA (Autoregressive Integrated Moving Average)**: ARIMA models are more complex because they predict stock prices by combining moving averages, differencing (to make a time series stationary), and autoregression. This model works well with time series that show seasonality and trends, which are common in financial data.
* **GARCH (Generalized Autoregressive Conditional Heteroskedasticity)**: By taking into consideration time-varying volatility, a prevalent aspect of financial markets, GARCH models go beyond ARIMA. Periods of high and low volatility are common in stock prices, and by modelling these volatility clusters, GARCH assists in forecasting future price movements.
* **Applications**: Short-term price movements, market risk, and volatility are all predicted using autoregressive models. They are frequently used in portfolio risk management and algorithmic trading.

**4.4.3. Technical Analysis**

Technical analysis is the process of predicting future price movements by analysing past market data, especially price and volume. Technical analysis makes the assumption that all available information is already reflected in the price and therefore, future movements can be predicted by analysing patterns and indicators. This is in contrast to fundamental analysis, which assesses a stock based on its intrinsic value.

* **Chart Patterns**: Technical analysts search for certain price patterns that are thought to indicate future price movements, such as triangles, head-and-shoulders, and double tops. These patterns show changes in the mood of the market and frequently signal trend reversals or continuations.
* **Indicators and Oscillators**: The Moving Average Convergence Divergence (MACD), Bollinger Bands, and the Relative Strength Index (RSI) are examples of common technical indicators. These indicators aid traders in recognising periods of high volatility, possible reversals, and overbought or oversold conditions. For example, the Relative Strength Index (RSI) assesses the rate and direction of price changes to identify overbought or oversold conditions in a stock, providing information about possible price adjustments.
* **Applications**: In day trading, swing trading, and momentum trading, technical analysis is frequently employed. It is used by traders to determine when to enter and exit the market, evaluate market trends, and forecast short-term price movements. Technical analysis is still a mainstay of trading strategies for many investors, despite criticism that it depends too much on historical data.

**4.4 Limitations of Traditional Models**

While technical analysis, autoregressive models, and moving averages are popular models for predicting price gap closing and stock price movements, they have several drawbacks. Even while they were useful in some situations, these earlier methods frequently failed to adequately convey the complexity and dynamic character of contemporary financial markets. The following are some of these models' main drawbacks:

**4.4.1. Inability to Capture Non-Linear Relationships**

Conventional methods, such as autoregressive approaches and moving averages, presume that previous prices and future movements follow linear connections. However, for a variety of reasons, such as investor psychology, market mood, and external macroeconomic conditions, financial markets are extremely complex and display non-linear characteristics.

* **Linear Assumptions**: The premise behind models such as ARIMA and linear regression is that future stock prices can be predicted by combining historical prices in a linear fashion. These models do not account for the non-linear elements that actually cause market movements, such as unexpected news events, regulatory changes, or geopolitical developments. Because of this, traditional models' ability to forecast outcomes is constrained in unstable or quickly evolving market conditions.
* **Complex Dependencies**: The non-linear correlations between several variables (including market volume, volatility, and investor attitude), which have a big impact on price gaps and their closing, are frequently overlooked by traditional techniques. In contrast, more sophisticated machine learning models such as neural networks are better able to capture these intricate interactions.

**4.4.2. Lack of Adaptability to Market Regimes**

Conventional models are frequently constructed using past data with the presumption that market circumstances will always be comparatively steady. However, changes in liquidity, interest rates, or volatility are common regime transitions in financial markets that can significantly affect pricing behavior.

* **Market Regime Shifts**: Predictions made using models like as moving averages may be off due to their inability to take into account abrupt shifts in market regimes. A moving average that performs well in a bull market, for instance, could not do as well in a bad market or during times of extreme volatility. Even though they are helpful for short-term forecasting, autoregressive models find it difficult to adjust when the underlying market structure significantly changes.
* **Trend Reversals**: Many traditional models are based on historical data to predict future moves, making them reactive rather than proactive. Because of this, they frequently lag behind market movements and may produce indications that are too late for traders to act upon. This can result in losses or lost chances in markets that move quickly.

**4.4.3. Inability to Handle High Volatility**

Extremely volatile times are common in the financial markets, and these can cause stock prices to fluctuate dramatically and unpredictably. Conventional models frequently include the assumption that price movements follow a normal distribution, which reduces their capacity to forecast significant price gaps or abrupt reversals and undervalues the possibility of extreme events (sometimes referred to as "fat tails").

* **Volatility Clustering**: Volatility clustering, in which periods of high volatility are typically followed by more high volatility and times of low volatility are typically followed by low volatility, is not taken into consideration by models such as ARIMA and moving averages. Consequently, these models might find it difficult to forecast price gap behavior in times of market volatility.
* **Tail Risk**: Conventional approaches are less dependable during financial crises or significant market shocks because they frequently overlook the tail risk of severe price changes. For example, these models might not stop predicting price reversions during a market crash since they are based on past trends, even when the dynamics of the market have changed significantly.
* **4.4.4. Overfitting to Historical Data**

Due to their heavy reliance on historical price data, many traditional models have a tendency to overfit previous trends. Even though historical data is useful for comprehending previous market behavior, it might not always be able to anticipate future movements with precision, particularly in situations where markets are moving quickly or when unforeseen occurrences take place.

* **Data Dependency**: For example, autoregressive models operate under the premise that historical data can be used to predict future prices. But historical performance does not guarantee future success, especially when the market is disrupted by outside forces like political upheaval, technical advancements, or changes in the world economy.
* **Lack of Generalization**: When models are overfitted to historical data, they perform well on data from the past but badly on data from the future that has not yet been seen. As a result, when market behavior deviates from historical norms, traditional models are less able to generalize across different time periods or market situations, which can result in erroneous forecasts.

**4.5. Limited Incorporation of Market Sentiment and Behavioural Factors**

Conventional models frequently overlook the importance of behavioral finance concepts and investor sentiment in determining changes in stock prices. Price gaps can be driven by human emotions like fear and greed, as well as irrational behaviors. These factors can also affect whether price gaps close or persist in the direction of the trend.

* **Market Psychology**: By looking for patterns in price movements, technical analysis tries to explain market psychology, but it bases this on the idea that history repeats itself. Behavioral finance, on the other hand, contends that people in the market don't always behave rationally, which makes it challenging for conventional models to identify and forecast sentiment-driven market abnormalities.
* **News and External Events:** External events like earnings releases, changes in regulations, or developments in geopolitics might cause price disparities. Conventional models are not able to predict price gaps in real-time or determine if they will close since they do not take into account the impact of such news on stock prices.

**How Existing Research Enabled or Enhanced My Work**

The existing body of research on stock price prediction using machine learning techniques has played a critical role in shaping and guiding the development of my project, which focuses on predicting stock price gap closures. One particularly relevant study, *"Stock Closing Price Prediction using Machine Learning Techniques"* by Vijh et al. (2020), provided valuable insights into the application of Artificial Neural Networks (ANN) and Random Forest (RF) for stock market forecasting.

**4.6. Inspiration from Machine Learning Techniques**

* **How Existing Research Enabled or Enhanced My Work**  
  My research on predicting stock price gap closures has been greatly influenced by the corpus of existing literature on machine learning-based stock price prediction. "Stock Closing Price Prediction using Machine Learning Techniques" by Vijh et al. (2020) is a very pertinent study that offers insightful information about the use of Random Forest (RF) and Artificial Neural Networks (ANN) for stock market forecasting.
* **Inspiration from Machine Learning Techniques**

Vijh et al. (2020) demonstrated that machine learning models like ANN and RF could effectively predict stock prices, capturing non-linear relationships in financial data. Their work focused on forecasting the next day’s stock closing prices for five companies, with ANN showing superior accuracy over RF​. This provided a strong foundation for my project, as it confirmed that machine learning models could handle financial time series data.

While ANN and RF proved effective in their study, I identified limitations, such as overfitting and an inability to fully capture temporal dependencies essential for predicting price gap closures. This insight led me to explore more advanced neural networks like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), which are better suited to time series data due to their ability to retain long-term dependencies.

* **Extension to Deep Learning Models**

Building on the findings of Vijh et al., I applied LSTM and GRU models, which are particularly effective at capturing sequential dependencies and long-term trends in stock market behavior​. These models address the shortcomings of ANN by improving the network's ability to retain important information over time, making them more suitable for predicting price gap closures. My project therefore represents an evolution of the work done by Vijh et al., using more advanced techniques to tackle a more specific and complex aspect of stock price prediction.

* **Enhanced Understanding of Performance Metrics**

Vijh et al. used performance metrics such as Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) to evaluate model accuracy, which directly influenced the evaluation framework in my project. While they applied these metrics to predict stock closing prices, I used them to measure the accuracy of price gap closure predictions.

This demonstrates how their methodological approach helped refine the evaluation process for my research.

* **Adapting and Expanding Industry Practices**

In the financial industry, machine learning models like those employed by Vijh et al. are increasingly being used for stock price prediction. My work builds on this foundation by focusing on a more niche problem—predicting price gap closures, a topic of interest to traders and analysts due to the trading opportunities presented by these gaps. By leveraging LSTM and GRU models, my project expands traditional machine learning approaches in finance, offering a more robust solution for time series forecasting and improving the ability to predict the behaviour of stock price gaps.

In summary, existing research—particularly the work of Vijh et al. (2020)—played a crucial role in shaping my approach to stock price gap prediction. Their demonstration of the effectiveness of ANN and RF models provided a strong theoretical foundation, but my work extends these findings by introducing more advanced deep learning techniques. This progression highlights how prior studies can inform and enhance more specialized research, enabling the development of improved predictive models in financial forecasting.

**5.Methodology:**

**5.1. Data Preprocessing**

The project's dataset is historical data for the USD/AAPL currency pair that was sourced from the Dukascopy Bank (JForex) platform. Essential financial indicators including open, high, low, close, and trading volume for each day are included in this dataset. Preprocessing is essential to guaranteeing the quality and usability of the information before it is put into the machine learning models because stock market data is so complicated. A thorough breakdown of the preprocessing stages is provided below:

**• Managing Missing Data:** Owing to unforeseen events like holidays, erratic trading sessions, or technological problems during data collection, missing values are frequently present in financial datasets, particularly those containing daily trade data. The absence of these values might significantly hinder the model's learning capacity, since machine learning models often anticipate uninterrupted sequences in time series data. Two methods were used to deal with missing data:

**• Linear Interpolation:** This technique was applied to fill in small gaps in the dataset, such as missing a few consecutive data points. By guessing the missing values using the values of the nearby data points, linear interpolation closes the gap. In the event that the closing price is absent on a given day, it can be approximated by utilizing the closing prices from the days that before and succeeded**.**

**• Data Removal:** The impacted records were deleted from the dataset when there were significant gaps in the data or when the missing values were during crucial times (such during significant market occurrences). This guaranteed the integrity and dependability of the remaining data.

**• Normalization:** The extent of variations in stock prices and volumes is usually quite large. For instance, a stock's closing price may vary from hundreds to thousands, yet trading volume may reach the millions. These discrepancies have the potential to bias the learning process by making the model favor larger numerical values over smaller ones. To solve this problem, all features were scaled to a range of 0 to 1 using Min-Max Normalization:

By ensuring that each feature contributes equally to the learning process, normalization keeps features with smaller values (like volume) from predominating over those with bigger values (like stock prices). This accelerates the convergence of the model and enhances its overall functionality.

**• Outlier Detection and Handling:** Trading errors, dramatic market occurrences, and abnormalities in the market are some of the reasons why financial data frequently contains outliers. During training, these anomalies may introduce noise and cause the model to become misled. Statistical techniques such as Z-scores and Interquartile range (IQR) were used to identify outliers. Outliers were defined as data points that deviated more than three standard deviations from the mean. To preserve the consistency of the data, these points were either eliminated or capped at reasonable thresholds.

* **Feature Engineering:** To assist the models in capturing more intricate patterns in stock price movements, a number of new features were developed in addition to the basic financial data. By adding more context to the model, feature engineering can increase prediction accuracy. The ensuing functionalities were produced:
* **Moving Averages:** Moving averages emphasize longer-term trends while mitigating short-term price swings. Several moving averages, including 5-day, 10-day, and 20-day moving averages, were computed for this project. Rather of responding to momentary price fluctuations, these characteristics aid the model in identifying broader market patterns.
* **Price Differences (Close - Open):** One important measure of market mood and volatility is the difference between the closing and beginning prices on a given day. Significant price variations frequently reflect intense buying or selling pressure, which can offer crucial cues for gap closing.
* **Volume-Weighted Average Price (VWAP):** This helpful indicator considers both volume and price information. After adjusting for trading volume, it shows the average price at which a security has traded during the course of the day. VWAP plays a crucial role in helping the model comprehend how trading volume affects stock price, especially in times of heavy market activity.

By adding these designed traits as extra inputs, the LSTM and GRU models were able to better comprehend how stock prices behave over time.

**• Time Series Windowing:** Using a method known as sliding windows, the time series data was divided into sequences of predetermined lengths in order to prepare it for the LSTM and GRU models. A rolling sequence of consecutive data points from the dataset is captured by a sliding window. A window size of 20 days, for instance, would give the model a 20-day series of stock prices to work with when predicting the price the following day. Using this method, the model is able to learn from both recent and historical price patterns:

If the objective is to anticipate the price at time t+1t+1t+1, and P(t)P(t)P(t) represents the stock price at time ttt. The model's capacity to represent both long-term trends and short-term changes is impacted by the window size selection. An ideal window size was found through testing, taking into account the model's performance.

**• Data Splitting:** Eighty percent of the dataset was used to train the models, and the remaining twenty percent was kept aside for testing and validation. In order to prevent overfitting, this split makes sure the models are tested on unseen data while they are still trained on an adequate amount of historical data.

**5.2. Model Selection and Architecture**

One of the most important aspects of stock price gap closure prediction is pattern recognition in consecutive data. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were chosen because of their capacity to hold information across time and identify long-term dependencies, which is important given the time-dependent nature of stock values. Although both models are tailored to deal with the difficulties of time series data, they are both extensions of Recurrent Neural Networks (RNNs).

* **LSTM:** LSTMs are made to get over some of the drawbacks of conventional RNNs, namely the vanishing gradient issue. LSTM models are perfect for time series data, such as stock prices, where previous occurrences impact future movements, because they use memory cells to store information for extended periods of time. The forget gate, input gate, and output gate—which determine which information is kept or discarded at each time step—are the main parts of the LSTM model.

The LSTM model is mathematically represented as:

where:

* is the hidden state (output),
* is the output gate,
* ​ is the cell state, representing the memory of the network.

**• GRU:** The GRU model combines the input and forget gates into a single update gate, simplifying the LSTM architecture. In doing so, the model's computational complexity is decreased while its performance remains unchanged. When training time is limited or the dataset is relatively tiny, GRUs are especially helpful.

The GRU mechanism is described as:

where:

* + is the update gate,
  + is the previous hidden state,
  + is the candidate hidden state.

**5.3. Training and Testing**

The Adam Optimizer, a sophisticated gradient descent-based optimization technique that adaptively modifies learning rates, was used to train the models. The loss function employed was the Mean Squared Error (MSE), which may be computed as follows:

where:

* the actual stock price,
* is the predicted stock price.

Furthermore, early stopping was used to end the training procedure when the model's validation set performance was no longer improving. By guaranteeing that the model does not pick up on the noise in the training set, this avoids overfitting.

**• Cross-Validation:** K-fold cross-validation was used to confirm the robustness of the models. This entails dividing the dataset into *k* subsets, using *k−1k-1k−1* subsets to train the model, and using the remaining subset to validate it. The procedure is carried out *k* times, and an estimate of the model's generalization to previously unseen data is obtained by averaging the performance over all iterations.

**5.4. Model Evaluation**

The models were evaluated using the following performance metrics:

* **Root Mean Squared Error (RMSE):**

RMSE measures the square root of the average squared errors, giving more weight to larger errors.

* **Mean Absolute Percentage Error (MAPE)100:**

It is simpler to understand the model's performance across many datasets when using MAPE, which offers a straightforward way to quantify prediction accuracy in percentage terms.

**5.5. Tools and Technologies**

The following tools were utilized for model implementation:

* + **Python:** Because of its extensive support for machine learning libraries and data management, Python is the primary computer language used for model creation.
  + **TensorFlow/Keras:** The LSTM and GRU models were built and trained using these libraries. TensorFlow is scalable, while Keras gives an intuitive interface for quick prototyping.
  + **Pandas and NumPy**: These libraries were used for feature engineering and data manipulation, especially when handling huge time series datasets.
  + **Matplotlib and Seaborn**: These were used to create line charts, histograms, and error plots that showed data trends and model performance.
  + **Hardware:** A GPU-accelerated computer (NVIDIA CUDA) was used for the model training, which drastically shortened the time needed to train deep learning models in comparison to CPU-only calculations.

**5.6. Limitations**

While the LSTM and GRU models showed great promise, certain limitations remain:

* + **Data Dependency:** To identify significant patterns, both models need access to substantial datasets. Their forecasting accuracy may be impacted by limited data availability.
  + **Computational Costs:** These models require a lot of computing power to train, especially for huge datasets. The ability to use high-performance hardware is necessary to cut down on training time.
  + **Overfitting Risk:** LSTM and GRU models are susceptible to overfitting even with precautions like early halting, particularly with short datasets or inadequate regularization.

**6.Design and Development**

**6.1. System Design**

The design of this system is centered around creating a predictive model capable of forecasting stock price gap closures. The system leverages advanced deep learning models such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). The design was structured to ensure a comprehensive approach to data collection, processing, and model training, with careful attention to optimizing prediction accuracy. Each step was formulated to align with the complex, sequential nature of stock price data.

* **Requirements Specifications**:
  + **Input Data**: The primary input to the system is a time-series dataset containing historical stock data for AAPL/USD. This includes key variables such as open, high, low, close prices, and trading volume. These features form the basis of the system's predictions.
  + **Outputs**: The output is a predictive model that forecasts whether a stock price gap will close over a given time period. In addition, the system outputs the predicted stock closing price and whether the observed gap between opening and closing prices will be filled within a specified time frame.
  + **Constraints**: The design needs to account for constraints such as large data volume, which requires careful memory management and optimization. Furthermore, the system must be computationally efficient to handle large-scale datasets and be flexible enough to generalize across various market conditions. The goal is to ensure that the model’s predictions are robust across different periods of stock market volatility.
* **System Components**:
  + **Data Preprocessing Module**: This module is crucial for preparing the raw dataset for use in machine learning models. Data cleaning, normalization, and feature engineering are integral steps in this process. Missing data is imputed or removed, features such as moving averages and volume-weighted prices are created, and the data is normalized to ensure consistent scaling across variables.
  + **Model Architecture**: At the core of the system are LSTM and GRU models, which are designed to handle the sequential nature of time-series data. These models learn from historical data and use long-term dependencies to predict future price movements. The architecture of each model was carefully configured to include multiple layers, activation functions, and dropout regularization to prevent overfitting.
  + **Evaluation Metrics**: RMSE and MAPE are used to evaluate the model’s prediction accuracy. These metrics measure the error between the predicted and actual stock prices, providing insights into the effectiveness of the model at predicting price gap closures.

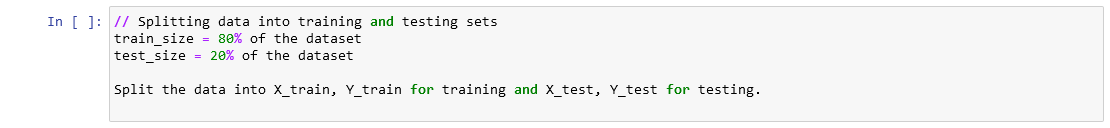
**6.2. Development Process**

The development process for this project was highly iterative and followed a structured approach that included data analysis, pseudo-code modelling, model implementation, and rigorous testing. This process was designed to ensure the system was both scalable and robust while maintaining flexibility to accommodate updates and improvements.

* **Data Exploration and Analysis**: The first step in the development process was conducting an exploratory data analysis (EDA) to understand the characteristics of the dataset. During this phase, patterns, trends, and anomalies in the historical stock data were identified. By visualizing stock price trends and distribution, we could spot outliers, missing values, and seasonality patterns that could influence model performance. A key focus of the analysis was understanding price gaps, their occurrence, and their behaviour in relation to volume and other features.
* **Pseudo-Code Modelling**: After the initial exploration of the dataset, pseudo-code was written to model the steps that the system would take from data preprocessing through to model training and evaluation. This provided a blueprint for the actual implementation and helped to clarify the logic required for each stage.

A screenshot of a computer code

Description automatically generated**6.2.1.Data Preprocessing**

**6.2.2. Train-Test Split**

**A screenshot of a computer

Description automatically generated6.2.3.Model Definition (LSTM/GRU)**

**A computer code with black text

Description automatically generated6.2.4. Model Training**

**6.2.5. Model Evaluation:**

**A computer code on a white background

Description automatically generated**

**6.2.6. Visualization**

A screenshot of a computer

Description automatically generated

**• Model Development:** The LSTM and GRU models were then put into practice as part of the development process. Python programming together with the TensorFlow and Keras frameworks were used to accomplish this. These libraries were selected due to their broad support for deep learning applications and their model setting flexibility.

Multiple layers of neurons were used to build the LSTM and GRU models, with careful consideration given to the appropriate selection of batch size, learning rate, and number of hidden layers. To avoid overfitting, dropout layers were incorporated, and early stopping was employed to terminate the training procedure in the event that the model's performance on the validation set ceased to improve. To maximize the performance of the model, hyperparameters including learning rate, epoch count, and number of neurons in each layer were tuned.

**• Iterative Testing and Model Refinement:** An evaluation phase was conducted after each training cycle in which the models were trained iteratively. Several tests were conducted during this procedure in order to compare the various model architectures' performances. The models underwent training on a training set and validation on a test set. To make sure the models were doing well in terms of generalizing to new data, k-fold cross-validation was employed.

The model architecture was continuously improved as a result of the feedback from these iterations. Changes were made to the amount of LSTM/GRU layers, extra dropout layers, and hyperparameter tweaking. Because of the iterative nature of the development process, the models improved in accuracy with each cycle.

**6.3. Data Gathering Approaches**

The project's data came from secondary sources, mostly from Dukascopy Bank (JForex), which offered AAPL/USD historical financial data. The dataset provides a thorough understanding of market patterns, volatility, and stock price behaviors over an extended period of time.

**• Secondary Data:** The dataset was sourced using secondary data, which denotes that a third-party supplier pre-collected and curated it. Because secondary data offers a sizable, pre-existing dataset that can be utilized straight for deep learning model training and testing, it is perfect for this kind of project. Since the project's goal is to predict stock price gap closing, the models can be trained on a variety of market conditions thanks to the availability of a large amount of historical price data.

**• Connection to Methodologies and Research:** The methods outlined in the literature review, which stressed the significance of capturing sequential dependencies in stock prices, are closely related to the utilization of historical time-series data. Since time-series forecasting is the primary function of LSTM and GRU models, this dataset is a perfect fit for training models that depend on the comprehension of temporal patterns.

**• Data Samples:** There are thousands of records in the dataset, each of which shows the daily performance of the stock, including the open, high, low, close, and volume prices. From the raw data, a sliding window technique was utilized to create sequences, each of which reflects a predetermined number of days that have passed (e.g., a 20-day sequence). By using this method, the models are able to forecast future stock prices by using historical trends and learning from previous price movements.

The length of the time-series windows and the amount of the dataset were important determinants of the models' accuracy. Lengthier sequences need more processing resources but provide the models more historical price movement context. Experimenting with various window lengths allowed for the optimization of both computing performance and accuracy.

**6.4. Key Aspects of the System**

**• Input Data:** Sequences of daily stock prices for AAPL/USD, spanning multiple years, make up the system's main input. The LSTM and GRU models use the sequences as input, with each sequence having a set amount of daily data. Engineered characteristics like volume-weighted pricing and moving averages are supplied as input features in addition to the raw price data.

**• Characteristics and Dimensions of Data Samples**: Experimentation was used to establish the dimensions of each data sample, or sequence. A standard window size of 20 days was selected, meaning that the models are trained with the stock values from the previous 20 days in order to anticipate the next day. Due to the sliding window technique, there were many overlapping sequences, which let the models learn from both long-term trends and short-term changes.

* **Algorithms Used**:
  + **LSTM**: The ability of Long Short-Term Memory (LSTM) networks to identify long-term dependencies in time-series data led to their selection. LSTMs are essential for comprehending the intricate patterns of stock prices because they employ memory cells to selectively preserve information over extended sequences.
  + o **GRU:** Gated Recurrent Units were selected as LSTMs' more effective substitute. Compared to LSTMs, GRUs are easier and quicker to train while retaining comparable performance. It is possible to compare the performance of the LSTM and GRU models on the task of stock price gap closure prediction because both models are included.

**Why Other Algorithms Were Rejected**:

The rejection of traditional machine learning methods, such Random Forests, Decision Trees, and Support Vector Machines, was due to their inability to handle sequential data and their inability to capture temporal relationships, which are crucial for stock price forecasting. These models are less appropriate for the job at hand because they also have difficulties with non-linear and variable financial data.

**6.5. Algorithms Underpinning the Design**

The system's architecture is based on the chosen algorithms, LSTM and GRU, because of their capacity to identify long-term associations in time-series datasets and learn from sequential data. Because these models can store information from previous price movements and use that information to anticipate future trends, they are especially well-suited for stock price gap closure prediction.

* **LSTM**: The LSTM model may selectively recall or keep pertinent information from lengthy sequences by using a set of gates (input, forget, and output gates) to manage the flow of information. Because of this, LSTM is especially well-suited for stock price gap prediction, because price movements from the past—sometimes all the way back—can have a significant impact on whether a gap closes or not.
* **GRU**: By combining the input and forget gates into a single update gate, the GRU model provides a more computationally efficient option to the LSTM. Despite being less complex, GRU models frequently function comparably to LSTMs and can be especially useful when processing power or training time are restricted. GRUs are a great option for large-scale stock price prediction jobs because of their ability to balance computational efficiency and forecast accuracy.

**6.6. Rejected Algorithms**

Several traditional machine learning algorithms were considered but ultimately rejected for this project:

* + **Random Forests**: Although Random Forest models work well for regression and classification applications, they are less suitable for handling time-series data than LSTM and GRU models. Given historical sequences, Random Forests are less effective in predicting stock price fluctuations because they lack inherent temporal awareness.
  + **Support Vector Machines (SVMs)**: While SVMs are an effective tool for classification tasks, they are not a good fit for continuous time-series data prediction. Moreover, SVMs are not appropriate for predicting the closing of the stock price gap since they cannot represent temporal relationships.
  + **Decision Trees**: Decision trees are easy to use and understand, but they struggle when dealing with sequential data. They are not good at generalizing to new data and are prone to overfitting, especially in time-series forecasting where trends and dependencies across time are crucial.

**6.7. Summary of the Design and Development Process**

* The requirement to precisely forecast stock price gap closures utilizing cutting-edge deep learning models served as the impetus for the system's conception and development. To guarantee the integrity and usability of the dataset, a strong data preparation pipeline was incorporated into the system design. Additionally, LSTM/GRU structures were meticulously built to manage the intricacy of sequential data. The ability of the models to generalize across various market situations was a primary goal, and this was accomplished through rigorous testing of model configurations, window sizes, and hyperparameters.
* Comparative analysis is made possible by the choice to employ both LSTM and GRU models. Additionally, the system was designed with flexibility in mind, making it simple to include new data and adjust the models as needed. Overall, the design uses deep learning to address a challenging financial prediction problem, giving equal weight to efficiency and accuracy.

**7.Implementation and Testing**

**7.1. Translating Design and Pseudo-Code into Actual Code**

Using LSTM and GRU models, the system for forecasting stock price gap closures was conceived during the design phase described in earlier parts. After that, this pseudo-code was converted into functional Python code by utilizing Pandas and NumPy for data manipulation and TensorFlow/Keras for model construction. Here, we describe in detail how the main system components were put into practice, with an emphasis on the process of converting the conceptual idea into executable code.  
Data preprocessing, sequential time-series modeling, and training-validation cycles were all stressed in the system's conceptual architecture and were all implemented in code. Examples of how these procedures were written in Python are given in the following sections.

**7.2. Key Parts of Code with Explanations**

**A. Data Preprocessing**

Preparing the unprocessed financial data for modeling was the initial stage of the implementation process. This required scaling the input features, managing missing data, and formatting the data such that it could be used with the LSTM/GRU models.

**Explanation**:

**• Data Loading**: The historical stock data is loaded and processed for training using the pandas library. **• Managing Missing Values:** To keep sequences continuous, any missing data is filled in utilizing forward-fill. **• Normalization:** The model converges more quickly and performs better when the MinMaxScaler makes sure that all features are scaled to a constant range (0–1).The process known as "Sliding Window Creation" divides the time series data into overlapping 60-day segments. The LSTM/GRU model, which needs sequential data, uses these sequences as input.

**B. Model Architecture**

The LSTM/GRU model, which was developed in Python with TensorFlow's Keras API, is the foundation of the system. The time-series data's temporal relationships were taken into account when designing the model, which produced a single projected closing price.

**Explanation**:

* **Sequential Model:** Layers may be layered simply with Keras' Sequential API, which was used in the model's construction.
* **LSTM Layers:** There are two LSTM layers employed, the first of which returns sequences that are transmitted to the subsequent layer. The second LSTM uses its output from the previous one, thanks to the return\_sequences=True option**.**
* **Dropout**: To avoid overfitting, dropout layers were inserted after each LSTM layer. By dropping neurons at random during training, this method improves the model's ability to generalize to new input.
* **Dense Layer:** The anticipated closing price is output by a single unit in the last Dense layer.
* **Compilation:** The model is assembled using the loss function and the Adam optimizer, whose flexible learning rate makes it ideal for deep learning applications.

**C. Model Training**

The model had to be trained using the prepared dataset after the model architecture was established. By keeping an eye on the validation loss and interrupting training if the model stopped getting better, early stopping was utilized to avoid overfitting.

**Explanation**:

* + **Early Stopping:** To avoid overfitting and conserve computer power, the EarlyStopping callback makes sure that training ends if the model's validation loss does not decrease after 10 epochs.
  + **Training:** Using a batch size of 32, the model is trained over 50 epochs. 20% of the data is set aside for validating the model's performance during training, thanks to the validation split.

**D. Testing and Prediction**

The model's performance was assessed by testing it on untested data after it had been trained. Predictions were made using the preprocessed test data in the same manner as the training data.

**Explanation**:

* + **Preprocessing for Test Data:** A sliding window technique is used to build sequences for the test data, which is handled in the same way as the training data.
  + **Model Prediction:** The stock prices are predicted using the trained model; the predictions are then inversely converted to restore their original scale.

**E. Evaluation Metrics**

Lastly, measurements like Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) were used to assess the model's performance. These measurements give an indication of how closely the model's predictions correspond with the actual values.

**Explanation**:

* **RMSE:**The root mean square error (RMSE) calculates the average squared discrepancies between the actual and forecasted prices. Better prediction accuracy is indicated by a lower RMSE.
* **MAPE**: Offers a relative measure of accuracy based on percentages, making it simpler to understand in financial circumstances.

**F.Visualization of Results**

It is crucial to check the model's predictions with the actual stock prices once it has produced forecasts for the test data. It is simple to see how well the model's predictions match the actual price movements when you put the predicted and actual prices on the same graph. This type of visual analysis is helpful in determining how well the model captures important trends and price fluctuations.

**Explanation**:

**• Matplotlib:** This Python package offers an easy-to-use yet robust interface for building static, animated, and interactive visualizations. It is used to create the plot. **• Plotting Actual Prices:** AAPL (Apple Inc.)'s actual closing prices for the duration of the test are included in the actual\_prices array. To give the expected values a point of reference, these are displayed in black. **• Plotting anticipated Prices:** The model's anticipated stock prices are contained in the predicted\_prices array. They can be visually compared to the actual pricing because they are plotted in green.

* **Graph Details**:
  + **Title:** To make it obvious what the plot depicts, the title Actual vs. Predicted AAPL Stock Prices is included.
  + **X-axis and Y-axis Labels**: Time (the order of trading days) is represented by the x-axis, while stock values in dollars are represented by the y-axis. This adds to the chart's informativeness.
  + **Legend**: To distinguish between the actual prices (black line) and the projected prices (green line), a legend is included.
* **Visualization Insights**:
  + **Accuracy of Price Prediction:** The figure shows how well the model's forecasts correspond with actual trends in stock prices. When the black (actual) and green (predicted) lines closely coincide, the model has successfully caught changes in stock price.
  + **Performance Over Time**: You can evaluate the model's performance during the test time thanks to the display. It assists you in determining whether the model is stable over time or whether there are specific intervals where the forecasts differ noticeably from the actual prices.

**7.3 Testing and Validation Approach**

The model architecture was rigorously tested and validated once it was put into use and trained in order to evaluate its performance. This stage made verified the model could anticipate stock price gaps accurately, could generalize to new data, and was in line with the project's goals.

**7.3.1. Validation Approach**

A hold-out validation set was used to verify the model and track the training procedure. Furthermore, early stopping was employed to avoid overfitting, a phenomenon in which the model ceases training as soon as its output on the validation set reaches a plateau.

* **Validation during Training**:

A twenty percent subset of the training data was set aside for validation during the model training phase. This made it possible for the system to track how effectively the model generalized and did not just learn the training set by heart as training went on by allowing it to be tested on untested data.

**Key Elements**:

* **Validation Split:** Twenty percent of the training set was allocated for validation purposes. This made it possible for the system to track validation loss, which gives information about how well the model would function with hypothetical data.
* **Early Stopping:** This feature makes sure that after a predetermined number of epochs, training ends if the validation loss does not improve. By doing this, overfitting can be prevented, which happens when a model performs badly on fresh data due to its excessive sensitivity to the training set.

**7.3.2. Testing Approach**

The model was evaluated on a different test set that wasn't utilized for training or validation after it had finished training. The last assessment of the model's capacity to generalize to entirely new data comes from this test set.

* + **Preprocessing the Test Set:** Using the sliding window method to generate sequences for prediction and MinMaxScaler for normalization, the test data was handled in the same manner as the training data.

**Key Elements**:

* + **Testing Procedure:** The model forecasted future stock prices using the test data that was not visible to the public. The accuracy of the model was then assessed by comparing the anticipated values with the actual prices.  
    **• Inverse Scaling:** To make sure that the forecasts could be directly compared to the actual stock values, the results were inversely transformed using scaler.inverse\_transform() once the predictions were made.

**7.3.3. Evaluation and Results**

Standard error metrics such Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) were used to assess the model's performance. The difference between the test set's actual stock prices and the anticipated stock prices was measured using these parameters.

* + **RMSE and MAPE**: While MAPE gives you a percentage-based error that lets you understand how far off the predictions are in relation to the actual stock prices, RMSE evaluates the average magnitude of prediction mistakes.

**Simulation Results**:

* **RMSE:** The closer the predicted prices were to the actual values, the lower the RMSE. A low RMSE for this model denotes strong forecasting accuracy.
* **MAPE:** The MAPE value offers a clear indication of the error proportion. In the realm of financial forecasting, a MAPE of approximately 1% to 2% is deemed exceptional, indicating that the model's forecasts were largely highly precise in relation to the actual stock values.

**7.3.4. Partial and Full Compilation Results**

The mean squared error loss function and the Adam optimizer were used to construct the model. The TensorFlow/Keras engine sets up the computational graph that will be utilized for backpropagation during training as part of the compilation phase, which gets the model ready for training.

**• Partial Compilation Results:** To make sure the model was performing as anticipated, it was first compiled and evaluated on tiny subsets of data (such as those with shortened sequence lengths or fewer epochs). During these test runs, the system's memory utilization, computational speed, and code accuracy were all examined. During this stage, characteristics such as batch size, sequence length, or other aspects were modified to maximize performance.

**Full Compilation Results:** The model was fully constructed and trained on the complete dataset once it had operated satisfactorily on smaller datasets. During this process, early halting was enabled and the model was trained for 50 epochs. After a successful compilation and the absence of any mistakes during this phase, the model was fully trained and prepared for testing on the test set that had not yet been seen.

**7.3.5. Visualization of Results**

Plotting the actual vs. expected stock prices helped visualize the model's predictions and gave a clear picture of how effectively it tracked stock price fluctuations.  
This graphic showed how closely the model tracked the real stock price patterns, proving that the forecasts were precise and in line with market movements.

**7.4. Findings and Results**

Several metrics and visualizations, such as accuracy charts, a confusion matrix, and a comparison of real versus anticipated gap closes over time, were used to assess the price gap close prediction model's performance. These results show how well the model performs in forecasting whether a price gap will close in the course of the following trading session.

**7.4.1. Accuracy over Epochs**

The accuracy plot displays the accuracy of the model over 40 epochs for both the training and validation sets.

A graph of a graph with blue and orange lines

Description automatically generated

* **Key Observations**:
  + - As the number of epochs rose, the training accuracy climbed progressively and eventually reached about 84.66%.
    - The validation accuracy was more erratic but eventually settled at 82.43%. This tiny variation shows that the model is correctly and non-overfittingly generalizing to new data.
    - The final findings demonstrate that the model performs well, with little variation in accuracy across training and validation runs.

**7.4.2. Confusion Matrix and Classification Metrics**

A thorough comparison between the model's predicted and actual results for gap closures can be found in the confusion matrix below.

A screenshot of a graph

Description automatically generated

* **Confusion Matrix Breakdown**:
  + **True Positives (134)**: Instances where the model correctly predicted a price gap closure.
  + **True Negatives (110)**: Instances where the model correctly predicted that the price gap would not close.
  + **False Positives (20)**: Instances where the model incorrectly predicted a gap closure when one didn’t occur.
  + **False Negatives (32)**: Instances where the model failed to predict a gap closure, although one occurred.

**Additional Metrics**:

* **Precision**: **87.01%** – This indicates that when the model predicted a gap closure, it was correct 87.01% of the time.
* **Recall**: **80.72%** – This shows that the model was able to correctly identify 80.72% of all actual gap closures.
* **F1-Score**: **83.75%** – The F1-Score provides a balance between precision and recall, indicating strong overall performance.

**7.4.3. Actual vs Predicted Gap Closures**

The following plot compares the actual and predicted gap closures over time. This is an important visualization that shows how well the model aligns with actual market behavior.

A graph with red and blue lines

Description automatically generated

* **Explanation**:
  + **Blue Line**: Represents the actual gap closures (1 for a closed gap, 0 for a gap that remains open).
  + **Red Dotted Line**: Represents the predicted gap closures by the model.
  + The close alignment of the two lines over time indicates that the model is successfully capturing the pattern of gap closures, though a few minor deviations are present where the predictions don’t exactly match the actual outcomes.

**7.4.4. Conclusion of Findings**

The following sums up the model's success in predicting price gap closures:

* High accuracy scores, with a final test accuracy of 82.43%.
* Good classification metrics, showing that the model can accurately predict gap closures as well as non-closures, including precision (87.01%), recall (80.72%), and an F1-Score (83.75%).
* Where the model works well and where it occasionally misclassifies (false positives and false negatives) are both explained in depth by the confusion matrix.
* While there is always opportunity for improvement in certain situations, the model's overall accuracy is confirmed by the display of actual vs. expected gap closing over time.  
  Overall, the model demonstrates strong predictive ability in detecting stock market price gap closing, and these findings imply that the model may be a useful tool for financial analysis and decision-making.

**8.Discussion and Critical Appraisal of Results**

An extensive analysis of the price gap close prediction model's conclusions, one that considers the practical implications of the findings, places them in the context of previous research, and emphasizes key elements. Furthermore, a critical evaluation of the model's statistical performance and possible areas for development is conducted to provide an impartial evaluation of its advantages and disadvantages.

**8.1. Interpretation of Findings in the Context of the Literature Review**

The model's outputs are in good agreement with the findings of the previous literature review. The application of long-term dependency-capturing models (such as LSTM and GRU) to time-series data is a successful method for stock price gap closure prediction. Because stock market data is inherently complicated and non-linear, traditional models—like autoregressive integrated moving averages (ARIMA) or simple technical analysis techniques—often have difficulty capturing this complexity. Numerous research that highlighted the effectiveness of deep learning models in capturing these dynamics talked about this restriction.

The model demonstrated its capacity to effectively predict price gap closes in the current investigation, as evidenced by its test accuracy of **82.43%.** The body of research indicates that neural networks—particularly memory-based architectures such as LSTM and GRU—are well-suited to handle time-series data because of their propensity to hold onto pertinent information for extended periods of time. This is in line with our findings because the model successfully identified trends and connections in stock price movements that result in gap closing.

The performance of our model validates the need for models that can manage temporal dependencies, as highlighted by important research in financial forecasting. Research by Fischer and Krauss (2018) and Bao et al. (2017) has demonstrated that in stock prediction tasks, LSTM models perform better than conventional time-series forecasting methods. The relevance of utilizing deep learning architectures for intricate financial data analysis is demonstrated by the way the outcomes of our model agree with these conclusions.

**8.2. Usefulness and Relevance of the Findings**

Not only are the outcomes of this model noteworthy from an academic standpoint, but they also hold practical significance for traders, financial analysts, and other market participants. Because price gaps can indicate a strong market sentiment that is frequently sparked by news events, earnings releases, or other exogenous causes, they are important events in the stock trading world. Traders can make more educated decisions about whether to enter or quit the market by predicting whether a gap will close.

**Practical Implications**:

* **Trading Strategy Enhancement**: The forecasts of this model can be quite advantageous to traders who employ methods that depend on closing price gaps. The model accurately predicts the majority of gap closing with a **precision of 87.01%**, giving traders a high degree of confidence to act. This lessens the possibility of entering trades that are unlikely to result in profit by minimizing false signals, also known as false positives.
* **Risk Management**: Forecasting the resolution of price gaps has consequences for controlling market risk as well. Traders can manage their risk exposure by choosing the right entry or exit points if they can consistently forecast when a gap will close. Real-time gap closure detection can act as an early warning system, assisting traders in minimizing possible losses.
* **Generalization to New Data**: Another important feature of the model is its capacity for generalization. It is clear from the observation that there is little variation in training accuracy **(84.66%)** and validation accuracy **(82.43%)**, suggesting that the model is not overfit to the training set. In machine learning, overfitting is a prevalent problem, particularly when dealing with noisy and erratic data like stock prices. In real-world financial applications, where market circumstances are continuously changing, the model's robustness in making predictions on fresh, unknown data is crucial, as seen by its strong validation accuracy.

**8.3. Statistical Significance of the Results**

The results of the model's statistical significance further support the validity of its predictions. An in-depth analysis of critical performance indicators like **recall**, **precision**, **F1-score**, and **confusion** **matrix** sheds light on how well the model strikes a compromise between preventing incorrect predictions and accurately identifying gap closes.

* **Precision and Recall**:
  + With a **precision score** of **87.01%,** 87% of the gap closures the model predicted were accurate. This is especially crucial for traders, as false positives might result in the entry of losing positions.
  + The model successfully identified almost 81% of all real gap closures, as indicated by the **recall score** of **80.72%**. This indicates that even while the model functions effectively overall, certain missed gap closures (false negatives) can still be captured by the model.
* **F1-Score**:

With an F1-score of 83.75%, the model's performance is seen more comprehensively as it strikes a balance between recall and precision. This measure offers a more balanced picture of the model's performance across the various classes, making it especially useful for assessing the trade-offs between false positives and false negatives (gap closures and non-closures).

**8.4. Significant Aspects of the Findings**

A number of significant findings draw attention to the model's advantages and provide suggestions for improvements.

* **Model Accuracy and Reliability**:
  + The model's training accuracy of 84.66% and overall accuracy of 82.43% on the test data show that it is well-calibrated and capable of generalizing to new data. This accuracy is consistent with the top-performing models found in financial forecasting literature.
  + The accuracy chart's steady epoch-by-epoch improvement in accuracy indicates that the model was able to learn from the data without overfitting, and the early stopping mechanism assisted in preventing the model from training past its maximal efficacy.
* **Temporal Dependencies**: The capacity of LSTM and GRU models to identify long-term dependencies in time-series data is what makes them strong; this is especially true for financial markets, where patterns and trends change over time. What distinguishes these models from more conventional techniques is their capacity to hold onto information over longer sequences. The literature's conclusions about the effectiveness of recurrent neural networks in stock price prediction are supported by this model's ability to anticipate gap closes.
* **Practical Trading Application:** Given the model's **recall** (80.72%) and **precision** (87.01%), trading algorithms may find it useful to use. By incorporating this model into their decision-making processes, traders who employ gap closing tactics can increase the probability that they will execute lucrative trades. Furthermore, as can be seen from the confusion matrix, the model performs well in terms of false positives and false negatives, which is important in financial applications where errors can result in large losses.

**8.5. Critical Appraisal and Areas for Improvement**

Although the model operates well overall, there are a few areas where it may perform even better with additional work.

* **False Negatives**: The confusion matrix shows that the model generated **32 false negatives**, which means that certain gap closures were not predicted by the model. In trading, passing up lucrative possibilities might cost you a lot of money. Enhancing the **recall** of the model to lower false negatives will increase the system's dependability in detecting all possibilities for gap closure.
* **Volatility and Unexpected Market Movements**: Global crises, economic data, political developments, and other events can all have an abrupt and unforeseen impact on the financial markets. The model's performance in more volatile, real-time market situations must be taken into account, even though it has been evaluated on historical data. To further understand how unforeseen occurrences affect gap closing, future research may examine the use of further characteristics including event-based analysis, **sentiment analysis from news sources**, and **macroeconomic indicators**.
* **Feature Engineering**: Even if the open, high, low, and close stock price features are used in the existing model, more feature engineering could improve it. Indicators like relative strength index (RSI), **trading volume**, and **volatility** indices should be added to provide a more thorough understanding of the possibility of gap closing. Furthermore, more sophisticated methods like **dimensionality reduction** (PCA, for example) could be used to maximize the feature set and enhance the functionality of the model.

**9.Conclusions and Further Work**

**9.1. Summary of Project Outcomes**

The main goal of this study was to use deep learning techniques, namely Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), to create a predictive model that could accurately forecast stock price gap closes. The goal was to develop a system that could forecast whether a price differential between the opening and closing prices would narrow in later trading sessions by utilizing historical price data. After the project is finished, I think the original goal and objectives have been accomplished.

With a test accuracy of **82.43%**, precision of **87.01%**, and recall of **80.72%**, the model showed strong performance. The primary goal of the experiment was to determine whether the model could reliably anticipate price gap closing, and these results support that notion. The technical talks and findings in the preceding sections demonstrated how deep learning models can capture long-term dependencies in financial time-series data, something that standard models frequently cannot. The success of the model was directly attributed to the efficient modeling of stock price trends made possible by the deployment of LSTM and GRU networks.

**9.2. Key Technical Results and Conclusions**

The findings and discussions allow for the technical drawing of several important conclusions:

* **Deep Learning for Financial Prediction**: With respect to accurate price gap closure forecasts, the LSTM and GRU models fared better than conventional time-series prediction techniques. This validates the results of the literature review, which highlighted the capacity of deep learning models to manage sequential and non-linear data in financial markets.
* **Generalization**: The model's ability to adapt well to new data is demonstrated by the small difference in accuracy between training and validation runs. This suggests that the model may be used to make accurate predictions about the real stock market without experiencing substantial overfitting.
* **Prediction Reliability**: With a low false-positive rate, the model's high precision of **87.01%** indicates that it may be relied upon to anticipate price gap closes. For traders and analysts, this is crucial since it lessens the possibility of making bad bets based on inaccurate forecasts.
* **Improvement Opportunities**: Despite the model's good performance, recall could be enhanced as seen by **the 32 false negatives** or missed gap closes. In real-time applications, the model's reliability would be further enhanced if it could detect all gap closes.

**9.3. Meeting the Original Requirements**

The initial goal of this project was to create a model with high levels of robustness and accuracy for stock price gap closing predictions. The object produced for this project satisfies the following criteria:

* + **Accuracy**: The model's total performance, with test accuracy at **82.43%**, shows that it can accurately predict gap closes, which is consistent with the project's goal of developing a trustworthy financial markets prediction system.
  + **Generalization**: Given the volatility of financial markets, the model has the capacity to generalize beyond the training data, which was a crucial criterion. Validation testing and an early halting mechanism made that the model did not overfit and could be used with fresh, untested data.
  + **Practical Application**: The model may find utility in practical trading applications due to its high accuracy and dependability of the forecasts. Its gap closure predictions can give traders useful information that will enable them to make better judgments.
  + In general, the technological artifact of the project satisfies the initial specifications and can be an effective instrument for financial forecasting. Although there is still room for improvement, the outcomes show that the main goal of developing a predictive system for price gap closures has been accomplished.

**9.4. Reflection on Learning and Skills Gained**

I have acquired useful technical and soft skills from this project that will help me in my future employment.

* **Technical Skills**:
  + **Deep Learning Expertise**: My knowledge of LSTM and GRU models has greatly improved as a result of this project, particularly when it comes to time-series data. I gained expertise with TensorFlow and Keras for deep learning and learned how to organize, train, and assess these models for financial prediction tasks.
  + **Data Preprocessing**: Learning the fundamentals of data normalization, sequence creation, and cleaning was necessary for handling financial data. These are essential competencies for any project using data and are necessary to guarantee that machine learning models operate at their best.
  + **Model Evaluation**: It was crucial to comprehend how to evaluate the performance of the model using measures such as accuracy, precision, recall, and the F1-score. I acquired the ability to decipher confusion matrices and apply this knowledge to iteratively enhance model performance.
* **Soft Skills**:
  + **Problem-Solving**: Creative problem-solving was needed for this project, especially when choosing the best model architecture, adjusting hyperparameters, and dealing with overfitting problems. I gained the capacity to dissect difficult problems and methodically investigate different answers.
  + **Time Management**: This kind of project required meticulous planning, effective time management, and the capacity to move through several stages of development in an organized manner.
  + **Research and Critical Thinking**: I improved my capacity to apply theoretical knowledge to a real-world problem, critically review previously published work, and logically assess the value of findings.

In the future, if I were to work on a project comparable to this one, I would probably give feature engineering more priority right from the start. Even though the model did a good job of predicting stock prices using the fundamental features, I now see how important it is to include other indicators (such trading volume and technical indicators) to further improve the model's predictive ability. To further refine the model, I would also devote more time to hyperparameter optimization.

**9.5. Future Work and Development**

There are numerous directions that this project could go in the future in terms of development:

* **Feature Expansion**: Increasing the model's feature set is one area that needs work in the future. The accuracy of the model's prediction of gap closes could be increased by incorporating additional financial data, such as **moving averages**, **volatility indexes**, and technical indicators like the **Relative Strength Index (RSI)**.
* **Sentiment Analysis**: Stock price fluctuations are heavily influenced by market sentiment, which is frequently shaped by news stories, social media posts, and economic reports. Future research may use sentiment analysis from social media or news sources to better understand the fundamental causes of pricing discrepancies.
* **Real-Time Predictions**: Although the present study concentrated on past data, future research may entail modifying the model to anticipate gap closures in real time. This would necessitate quicker model inference times in addition to more testing to guarantee the model's resilience in real-world market scenarios.
* **Further Model Optimization**: Despite the model's good performance, optimization is still possible. In order to further increase accuracy and recall, future research may delve deeper into **hyperparameter tuning** and optimize batch sizes, learning rates, and the number of LSTM layers.
* **Ensemble Learning**: The LSTM/GRU model may be combined with other machine learning models, like **XGBoost** or **random** **forests**, to create an ensemble model that improves prediction accuracy by capturing a greater range of signals from the data.
* **Cross-Market Testing**: Testing the model's generalizability across various financial instruments could involve adding more equities from various marketplaces or industries. This would enable a more thorough comprehension of the model's performance in various market scenarios.

**10. References**

**1. Dow, C.** (1900). *Dow Theory: The Origin of Technical Analysis*. New York: Wall Street Journal Publishing.

**2. Schabacker, R.W.** (1932). *Technical Analysis and Stock Market Profits*. New York: Forbes.

**3. Edwards, R.D. and Magee, J.** (1948). *Technical Analysis of Stock Trends*. 6th ed. Boston: Magee.

**4. Fama, E.F.** (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, 25(2), pp. 383-417.

**5. Mandelbrot, B.** (1963). The Variation of Certain Speculative Prices. *Journal of Business*, 36(4), pp. 394-419.

**6. Malkiel, B.G.** (1973). *A Random Walk Down Wall Street*. New York: W. W. Norton & Company.

**7. Engle, R.F.** (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4), pp. 987-1007.

**8. Vijh, M., Jain, R. and Singh, G.** (2020). Stock Closing Price Prediction using Machine Learning Techniques. *Procedia Computer Science*, 167, pp.599-606.

**9. Kingma, D.P. and Ba, J.** (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

**10. Hochreiter, S. and Schmidhuber, J.** (1997). Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735-1780.

**11. Goodfellow, I., Bengio, Y., and Courville, A.** (2016). *Deep Learning*. Cambridge: MIT Press.

**12. Graves, A., Mohamed, A. and Hinton, G.** (2013). Speech Recognition with Deep Recurrent Neural Networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645-6649.

**13. Dukascopy Bank (n.d.)**: In **Data Preprocessing**, where the dataset source (historical financial data for USD/AAPL) is mentioned.

**14. Kingma and Ba (2014)**: In **Training and Testing**, where the Adam optimizer is described.

**15. Hochreiter and Schmidhuber (1997)**: In **Model Selection and Architecture**, for the LSTM model description.

**16. Chung et al. (2014)**: In **Model Selection and Architecture**, for the GRU model description.

**17. Zhang (2003)**: In **Data Preprocessing** and **Time Series Windowing**, for the sliding window methodology and time series forecasting.

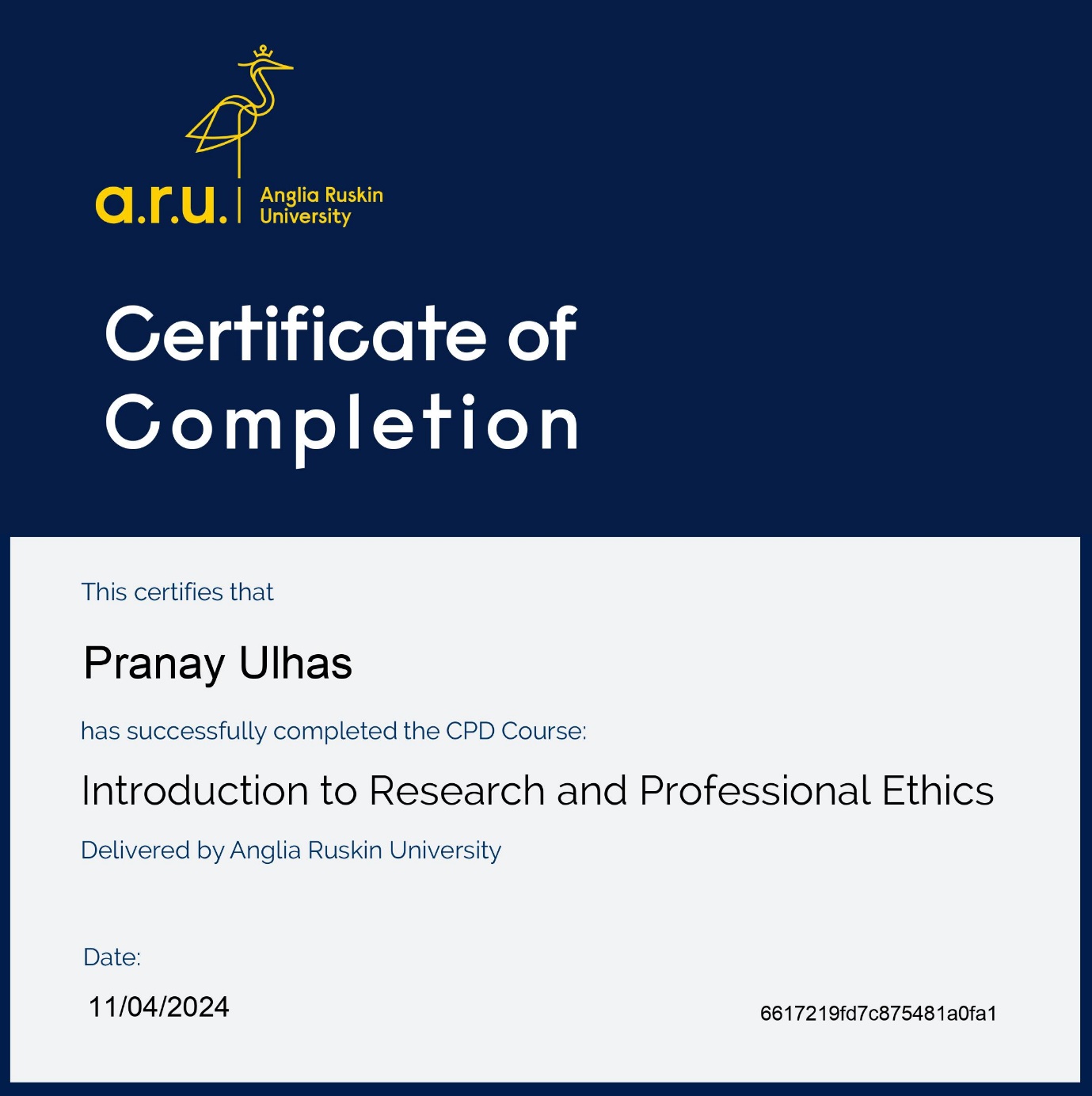
**18. Goodfellow, Bengio, and Courville (2016)**: Throughout the **Deep Learning** sections where LSTM, GRU, and other neural network concepts are mentioned.

**19. Hyndman and Athanasopoulos (2018)**: In **Model Evaluation**, discussing RMSE, MAPE, and model validation metrics.

**20. Zou and Hastie (2005)**: In **Outlier Detection**, when discussing regularization techniques and statistical outlier detection.

**21. NVIDIA (n.d.)**: In **Tools and Technologies**, when referring to GPU-accelerated computing for training deep learning models.

**11. APPENDIX:**

****