INTERNATIONAL
HELLENIC
UNIVERSITY

Stock Market Prediction using Double-DQN and Sentiment Analysis

Gkaimanis Dimitrios

SID: 3308210014

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2024

THESSALONIKI – GREECE

Stock Market Prediction using Double-DQN and Sentiment Analysis

Gkaimanis Dimitrios

SID: 3308210014

Supervisor:	Assoc. Prof. Christos Tjortjis
Supervising Committee Members:	Assoc. Prof. Paraskevas Koukaras
	Assist. Prof. Leonidas Akritidis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Information and Communication Systems

JANUARY 2024

THESSALONIKI – GREECE

Abstract

This dissertation was written as a part of the MSc in Data Science at the International Hellenic University. It presents a thorough investigation into the use of a Double Deep Q-Network reinforcement learning model, which leverages Technical Indicators and Sentiment Analysis extracted from the social media platform StockTwits, to forecast stock market trends. The case study in this thesis is NVIDIA stock, which demonstrated non-stationary similar behavior with a clear upward trend from January 2, 2020, to September 21, 2023.

Financial data for the analysis were sourced from Yahoo Finance, with technical indicators computed via the *yfinance* Python library. For the sentiment analysis, the *'Twitter-roBERTa-base for Sentiment Analysis'* model was employed, a tool that has been rigorously trained on roughly 58 million tweets and fine-tuned for sentiment classification using the *TweetEval* framework. Daily sentiment data were aggregated from StockTwits, capturing the market's pulse in response to news events, price changes, and general sentiment. The outcomes of this study are presented in the relevant chapter of this study and underscore the promising future of reinforcement learning models in stock market prediction, especially those that integrate sentiment analysis, indicating a transformative step forward for algorithmic trading approaches.

Acknowledgements

I would like to express my deepest gratitude to my thesis supervisor, Assoc. Prof. Christos Tjortjis, for his invaluable guidance, support, and insightful critiques throughout the duration of this research. His expertise and methodical approach to academic inquiry have significantly contributed to the depth and quality of this study. His encouragement and commitment to academic excellence have been a constant source of inspiration and have greatly facilitated my scholarly development.

Special thanks are also due to Vangelis Sarlis and Giorgios Papageorgiou, PhD candidates, whose perspectives and analytical skills have been instrumental in refining the research process. Their contributions were key for the successful completion of this thesis.

Gkaimanis Dimitrios

January 2024

Contents

Figures

1 Introduction

This thesis explores the innovative use of a Double Deep Q-Network (DDQN), a Reinforcement Learning model, for predicting stock market trends, specifically focusing on the NVIDIA stock. By integrating technical indicators and sentiment analysis derived from StockTwits tweets, this research aims to enhance the learning process and performance of the trading agent. Technical indicators provide a quantitative analysis of market trends, while sentiment analysis offers qualitative insights into investor behavior, together promising a more comprehensive understanding of market dynamics.

The hypothesis posits that these added layers of information will significantly improve the DDQN's predictive capabilities, leading to more informed and profitable trading strategies. This investigation not only sheds light on the practical applications of advanced machine learning techniques in the volatile world of stock trading but also explores the impact of investor sentiment and technical analysis on market predictions, providing valuable insights for both theoreticians and practitioners in the field.

1.1 Stock Market Forecasting

Stock market prediction has long been a subject of intense interest and study within the field of finance. The task of accurately forecasting the market is a matter of constant research and experimentation. Historically, a lot of methods have been employed in attempts to predict stock market trends. Among these are technical analysis, which relies on the study of past market data, primarily price and volume, and fundamental analysis, which evaluates securities by attempting to measure the intrinsic value of a stock [1]. Other methods include quantitative analysis, which employs mathematical and statistical modeling, and econometric models that analyze economic data to predict market trends. These traditional methods, while varied and sophisticated, have their limitations, especially in terms of handling the vast and complex datasets that characterize modern financial markets.

The emergence of machine learning models has created a new status into this field, offering novel approaches and tools to tackle the challenges of stock market prediction. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. In the context of stock market prediction, machine learning models are trained on vast arrays of historical data, much like human traders, but with the capacity to process and analyze data volumes far beyond human capability. This ability to digest and make sense of enormous datasets enables these models to uncover complex patterns and relationships that might elude human analysis.[2]

One of the newest applications of machine learning in finance is algorithmic trading. Algorithmic trading uses computer programs to trade at high speeds and volumes based on a set of predefined criteria and signals. These signals can be derived from a variety of sources, including market indicators, price patterns, or even news events. By automating the trading process, algorithmic trading allows for executing trades at the best possible prices, reducing transaction costs, and managing risk more effectively. Moreover, these algorithms can be designed to adapt and learn from market changes over time, continually refining and improving their trading strategies. [3]

In addition to the traditional data sources used in stock market prediction, the rise of social media platforms has opened a new limitless source of valuable information. Platforms like *Twitter*, *StockTwits*, and *Reddit* have become increasingly popular, providing a space where individuals and investors can express their opinions, expectations, and beliefs about various stocks and market trends. This user-generated content has become a rich source of sentiment data, capturing the mood, trends, and anticipations of people regarding specific stocks or the market as a whole.

The integration of sentiment analysis from social media into stock market prediction models is a growing area of interest. Sentiment analysis involves using natural language processing, text analysis, and computational linguistics to systematically identify, extract, quantify, and study affective states and subjective information. By analyzing the sentiment expressed in social media posts, researchers and analysts can gauge public opinion on a particular stock or the market in general. This sentiment data can be particularly useful in understanding and predicting short-term market movements, as public sentiment can often drive rapid changes in stock prices.[4] The combination of traditional market data with sentiment analysis from social media represents a holistic approach to stock market prediction. Machine learning models that are capable of processing and analyzing both these types of data can potentially provide more accurate and comprehensive market predictions. Such models can capture not only the historical trends and patterns in market data but also the current mood and expectations of the market participants.

The potential of machine learning in stock market prediction is vast. With the ability to process and learn from unlimited data and technical indicators, these models offer a level of analysis that surpasses human capability. They can identify subtle and complex patterns in the data, which might be invisible to the human eye. This capability can lead to more informed and effective trading strategies, potentially yielding higher returns. However, it is also important to recognize the challenges and limitations inherent in this approach. The stock market is influenced by numerous factors, many of which are unpredictable or unknown. Therefore, while machine learning models can significantly enhance the accuracy of stock market predictions, they cannot guarantee success.

1.2 Reinforcement Learning in Stock Market Prediction

Reinforcement Learning (RL) represents a paradigm shift in the landscape of machine learning and predictive modeling, particularly in dynamic and complex environments like the stock market. At its core, reinforcement learning is about decision making. It involves training an agent to make a sequence of decisions by interacting with an environment to achieve a certain goal. The agent learns to achieve this goal by trial and error, receiving feedback in the form of rewards or penalties for its actions. This learning process is akin to the way humans learn from their experiences.[5]

In the context of stock market prediction, an RL agent can be trained to decide on stock trading actions (buy, sell, hold) based on the current market state, with the objective of maximizing cumulative profit over time. Unlike supervised learning models, which require labeled data for training, RL can learn optimal strategies directly from the interaction with the market environment, making it particularly suitable for problems where the decision-making process is sequential, and the optimal strategy is not known in advance.[5]

Q-Learning, a form of model-free reinforcement learning, is a significant step in this direction. It involves learning a function, called the *Q-function*, which estimates the value of taking a particular action in a given state. This value estimation helps the agent to make decisions that maximize the expected return. *Q-Learning* has been successfully applied in various domains but faces challenges when dealing with high-dimensional spaces or continuous action spaces, common in stock market scenarios.[6]

The *Double Deep Q-Network* (Double DQN) model, an advancement over standard Q-Learning, addresses these challenges by using deep neural networks to approximate the Q-function. The 'double' aspect refers to the technique of using two networks to decouple the action selection from the target Q-value generation, reducing the overestimation bias present in traditional Q-Learning. This approach significantly enhances the model's ability to handle complex, high-dimensional environments like the stock market. [7]

Double DQN's capacity to process vast amounts of data and learn from a complex, ever-changing environment makes it a powerful tool for stock market prediction. It can process various types of data, including historical price data, technical indicators, and sentiment analysis from social media, to inform its trading decisions. By continuously interacting with the market and learning from its outcomes, a Double DQN model can potentially identify profitable trading opportunities and strategies that might be missed by traditional methods or human traders. [8]

However, applying RL and Double DQN models to stock market prediction is not without challenges. The stock market is a highly stochastic environment with a lot of noise and uncertainties, which can make the learning process difficult and the outcomes unpredictable. Additionally, the financial market's nature—mostly non-stationary and influenced by a plethora of external factors—poses a significant challenge for any model trying to learn and predict its behavior.[9]

Despite these challenges, the application of reinforcement learning, particularly Double DQN models, in stock market prediction represents a promising area of research. These models offer a novel approach to algorithmic trading, one that is dynamic, adaptive, and capable of processing complex, multidimensional data. As the field of machine learning continues to evolve, and as more sophisticated models and algorithms are developed, the potential for these techniques to revolutionize stock market prediction and trading strategies becomes increasingly apparent. The ongoing advancements in reinforcement learning, coupled with the growing availability of data and computational resources, are likely to lead to more robust, efficient, and profitable trading models in the future.

2 Literature Review

This chapter dives into a wide-ranging review of past work on predicting stock market trends, covering everything from basic economic ideas to the latest in computer methods. It lays down the groundwork that supports the research in this thesis. Starting with the main theories about how markets behave, the review moves through traditional methods and ends with a look at the latest machine learning techniques. It gives particular focus to recent developments in sentiment analysis, showing its increasing role in prediction models.

2.1 Principle Theorems of Stock Market Prediction

This section delves into the fundamental theories that have historically underpinned the understanding of stock market dynamics. It begins by discussing the Efficient Market Hypothesis, which posits that market prices reflect all available information and thus cannot be consistently predicted. In contrast, the Random Walk Theory suggests that stock prices are inherently unpredictable, moving randomly in response to unforeseen events. These theorems set the stage for the subsequent exploration of predictive methodologies, providing a theoretical backdrop against which the efficacy and limitations of various approaches can be measured. Understanding these foundational theories is crucial for anyone seeking to navigate the complex and often contradictory landscape of stock market prediction.

2.1.1 Efficient Market Hypothesis

The *Efficient Market Hypothesis* (EMH) has deeply influenced the realm of financial economics, significantly shaping how academics, professionals, and investors perceive market dynamics and the movement of stock prices. Originating in the 1960s from the work of economist Eugene Fama, EMH proposes that financial markets efficiently incorporate and reflect all available information within stock prices [10]. This proposition carries far-reaching implications for investment strategies, portfolio management, and the very essence of market speculation.

There are three main forms of the Efficient Market Hypothesis as Roberts H. [11] first introduced. These are the following:

- *Weak Form*: The weak form asserts that past stock price movements and trading volumes cannot predict future price changes, making technical analysis futile [12]. This implies that historical price data alone cannot provide an edge in forecasting future price movements.
- *Semi-Strong Form*: The semi-strong form posits that all publicly available information, including financial reports and news, is already incorporated into stock prices, rendering fundamental analysis ineffective [12]. Any new information is rapidly absorbed by the market, leaving little room for investors to gain an advantage based on publicly accessible data.
- *Strong Form*: The strong form contends that all information, whether public or private, is already reflected in stock prices, negating the possibility of any individual or group consistently outperforming the market [12]. Even private or insider information is assumed to be quickly factored into stock prices, making it challenging for investors to gain an unfair advantage.

The implications of EMH resonate beyond academics. For investors, understanding EMH guides investment strategies. Advocates argue that consistently outperforming the market is unlikely due to the swift incorporation of new information into prices. This aligns with the growth of passive investment strategies, which seek to replicate the overall market's performance [13].

Critics, however, challenge EMH's assumptions. Behavioral finance, an integration of psychology and economics, suggests that human biases and emotions can lead to market inefficiencies. This implies that not all information is accurately or instantaneously incorporated into prices, questioning the full efficiency of markets [13].

The Efficient Market Hypothesis stands as a cornerstone of modern finance and investment strategies. Its influence is evident in the ascendancy of index funds and passive investment. Yet, the ongoing dialogue between adherents and critics underscores the multifaceted nature of financial markets. The EMH serves as an essential framework for interpreting market dynamics, open to ongoing examination and refinement as the financial landscape evolves.

2.1.2 Random Walk Theory

The *Random Walk Theory* originates from the French Mathematician Louis Bachelier when firstly introduced the term in 1900 [14]. Following up his work, the theorem was made broadly known by economist Eugene Fama. This theory posits that stock prices follow an unpredictable and random pattern, rendering attempts to forecast future price movements ineffective. Central to the Random Walk Theory is the notion that stock prices swiftly absorb all available information, leaving no discernible pattern that can be exploited. In addition, the concept of random walks in stock prices encompasses two distinct ideas: firstly, that consecutive price alterations are unrelated; and secondly, that these price shifts adhere to a specific probability distribution. [15]

Malkiel's publication, "A Random Walk Down Wall Street," recently brought this theory into the mainstream. He used an analogy of a person taking random steps to illustrate the erratic nature of stock price movements, underscoring the challenge of consistently predicting market trends. Malkiel's work highlighted that actively managed funds often struggle to surpass passive index funds, which mirror overall market performance and align with the tenets of the Random Walk Theory [15].

More scientifically, according to Malkiel, a random walk refers to a scenario where future actions or directions cannot be foreseen based on past events. When this concept is applied to the stock market, it implies that short-term fluctuations in stock prices cannot be predicted. Services offering investment advice, predictions about earnings, and complex chart analyses prove ineffective. In the financial world, the term "random walk" is considered a rather derogatory expression. It was coined within academic circles and is used to refer to professional forecasters. Pushed to its logical conclusion, it suggests that even a blindfolded monkey randomly selecting stocks could achieve performance equal to that of portfolios crafted by experts.[15]

MackKinlay discusses the rejection of the random walk model and its implications for stock-price efficiency. He emphasizes that dismissing the model doesn't automatically indicate inefficiency in stock-price formation. Instead, the results set constraints on acceptable economic models for asset pricing. Any rational price formation framework must now account for the observed serial correlation pattern in weekly data. MackKinlay also highlights the utility of their specification test as a descriptive tool for studying price evolution over time. This is particularly valuable when prioritizing an empirically ground statistical model of the price process over a detailed economic equilibrium explanation. For instance, the pricing of complex financial instruments often hinges on understanding the specific stochastic process driving underlying asset returns, with less focus on the exact economic equilibrium generating prices. [16]

2.2 Methodologies of Stock Market Prediction

This section reviews a variety of established strategies used for anticipating stock market trends. It includes diverse techniques, ranging from detailed evaluations of company financials to statistical analyses of historical price movements. This part aims to shed light on how these various approaches have been applied over time, their relative advantages, and their roles in the broader context of market prediction. It provides an introductory understanding of the conventional tools and methods that analysts and investors have relied upon to forecast market directions.

2.2.1 Classical Approaches

This subsection delves into traditional methods commonly used to predict stock market trends. It starts with fundamental analysis, which deeply examines a company's financial state and market standing to assess its value. It then moves to technical analysis, a technique that studies past market data to spot patterns suggesting future price changes. Additionally, it explores time series analysis, using statistical tools to forecast a stock's future behavior based on its past. Finally, it mentions momentum investing, a strategy betting on the continuation of current market trends. Together, these classical methods provide a solid framework for understanding and anticipating stock market movements.

2.2.1.1 Fundamental analysis

Fundamental analysis can be perceived as a systematic approach with established principles and procedures aimed at determining the inherent value of a stock in the market. This approach operates within a broad framework that involves scrutinizing anticipated economic forecasts to identify sectors likely to experience growth in sales and profits. Subsequently, it involves a thorough financial evaluation of companies based on their historical financial data, current status, and the effectiveness of their management. It also assesses their potential business opportunities to ascertain the true value of their shares. This intrinsic value is then compared with market values, which are outcomes of demand and supply interactions, to identify potential investment opportunities, whether they may lead to profits or losses. [12][17][18]

Bauman's 1996 work suggests that the foundation of research in stock valuation via fundamental analysis can be attributed to three key papers. [19] The initial study by Lev and Ohlson in 1982 emphasized the importance of formulating models for stock valuation that could enhance and extend the existing correlation studies focused on the market within the realm of accounting research. [20]

The second influential paper was by Lev in 1989, which argued that our grasp of the process of analyzing financial statements hardly extends beyond a set of financial ratios that investors are assumed to use. This paper also highlighted the need for a shift in the focus of capital market research to address and evaluate key issues.[21]

The final paper, by Bernard in 1994, offered a critique of empirical research on the role of accounting data in stock valuation. Among his recommendations for future research was a call for a greater focus on modeling methodologies and studies that use samples of firms from specific industries or economic sectors. This approach would allow researchers to make better use of their detailed understanding of disclosures and institutional knowledge.[22]

Following these three pivotal studies, a wave of researchers began to explore the analytical and empirical relationships between accounting and non-accounting data in relation to firm equity valuation.

2.2.1.2 Technical Analysis

Technical analysis is a method used by analysts to forecast stock market trends by studying charts that display historical market prices and technical indicators.[23]

Technical analysts use historical stock data, which is structured data, and apply data preprocessing techniques. The processed data is then used in a predictive model to make market movement calls, such as buy, sell, or hold. Various technical indicators are utilized in technical analysis. Some of these, include the simple moving average (SMA), exponential moving average (EMA), moving average convergence/divergence rules (MACD), relative strength index (RSI), and on-balance-volume (OBV).[24][25][26]

According to Murphy (1999), the Technical Analysis approach relies on three fundamental principles [27]:

1. All market information is reflected in the price.
2. Price movements follow identifiable patterns or trends.
3. Historical patterns tend to recur over time.

The first principle is the foundation of Technical Analysis. Understanding and accepting the full implications of this principle is crucial for the rest of the analysis to make sense. Supporters believe that all factors that could potentially influence the price - whether they are fundamental, political, psychological, or otherwise - are already incorporated into the market price. Regarding the 2nd principle, acceptance of the fact that markets indeed exhibit trends is crucial before proceeding further. The primary goal of mapping market price action is to spot trends at their inception, to trade in alignment with these trends. Lastly, the 3rd principle takes into consideration that human psychology and stock market trading are interconnected. For instance, chart patterns, which have been recognized and classified over the past century, depict specific images on price charts. These images represent the market's optimistic or pessimistic psychology. Given their successful application in the past, it's assumed that they will continue to be effective in the future. These patterns are based on the study of human psychology, which tends to remain constant. The key to predicting the future lies in studying the past, or that the future is merely a repetition of past events.[27]

2.2.1.3 Timeseries Analysis

Time Series Analysis plays a crucial role in predicting stock movements, employing historical data to forecast future stock prices. This approach operates on the premise that past price actions and historical patterns can offer valuable insights into future market movements. Multiple methodologies and concepts contribute to this practice, drawing on insights from various sources.

A foundational aspect of Time Series Analysis involves the examination of trends and patterns in stock prices. For instance, technical analysis delves into historical price charts to identify patterns like moving averages, support and resistance levels, and various chart formations. These patterns are believed to hold clues regarding potential future price shifts [27].

Moreover, the utilization of *Autoregressive Integrated Moving Average* (ARIMA) models is prevalent in time series forecasting. ARIMA models consider a stock's historical price data, accounting for trends, seasonality, and fluctuations to make predictions about forthcoming price changes. Due to their efficacy in capturing intricate patterns in stock price data, these models are widely employed in financial forecasting [28].

Furthermore, the Efficient Market Hypothesis (EMH) plays a pivotal role in guiding time series analysis for stock prediction. EMH posits that stock prices already encompass all available information, making it challenging to predict future price movements solely based on historical data. This theory significantly influences the development and evaluation of time series models, emphasizing the efficiency of financial markets [10].

In summary, Time Series Analysis for stock prediction encompasses a range of techniques and principles, including technical analysis, ARIMA models, and machine learning approaches which will be discussed in depth in a following chapter. It draws insights from various quarters to scrutinize historical stock price data and provide forecasts for future price dynamics. While these methods offer valuable tools for investors and traders, it's crucial to consider the implications of the Efficient Market Hypothesis within the framework of time series analysis.

2.2.1.4 Momentum Investing

Momentum Investing holds a significant place in finance, suggesting that assets that have recently shown strong performance are likely to continue this trend in the short term, while underperforming assets will continue to struggle [29][30].

The heart of Momentum Investing is taking advantage of immediate price trends. Assets that have experienced positive price movements over a specific period, typically six to twelve months, are labeled as having positive momentum. Conversely, assets with negative price changes exhibit negative momentum. This contrasts with the efficient market idea, which assumes that all available information gets quickly absorbed into prices, leaving no room for profitable trends [29].

Momentum Investing's validity has been confirmed across various markets. A groundbreaking 1993 study by Jegadeesh and Titman showed that short-term past winners outperformed losers [29]. Further research by Moskowitz and Grinblatt (1999) and Chan, Jegadeesh, and Lakonishok (1996) underscores the prevalence of momentum across markets and its effect on returns [30][31].

Nevertheless, critics have raised valid concerns. The frequent trading associated with this strategy can result in high transaction costs. Skeptics argue that its success might be due to insufficient consideration of risk factors [30]. Additionally, some studies have highlighted momentum reversals, casting doubt on the idea of a consistently profitable strategy [31].

Momentum Investing also intersects with discussions in behavioral finance. Andreassen and Kraus (1990), made several significant observations in their study. In general, their findings indicated that participants tend to "track prices," meaning they sell when prices are rising and buy when prices are falling. This behavior persists even when the data they are provided with follows a random walk pattern, which aligns with the notion of underreacting to market news. However, when participants are presented with data that appears to have a discernible trend, they exhibit reduced tracking behavior, meaning they engage in fewer trading activities in response to price fluctuations. It remains unclear from Andreassen and Kraus's research whether participants transition from resisting trends to following them, but their results certainly suggest this possibility. [32]

Momentum Investing challenges established market theories by suggesting that past price trends can predict short-term future movements. Despite its empirical support and practical application, the theory faces skepticism regarding its limitations and behavioral aspects. Research conducted by Jegadeesh, Titman, Moskowitz, Grinblatt, Chan, Jegadeesh, and Lakonishok offers foundational insights into this pivotal theory. [29][30][31]

2.3 Machine Learning Approaches for Stock Market Prediction

This section introduces machine learning as a cutting-edge frontier in stock market prediction, highlighting its capability to uncover complex patterns and relationships within market data. It explores various sophisticated algorithms that have shown promise in forecasting market trends. Initially, it discusses Support Vector Machines (SVM), a powerful classification tool that's been adapted for market trend prediction. Following this, it examines Long Short-Term Memory Networks (LSTM), a type of neural network particularly adept at recognizing long-term dependencies in time-series data. Lastly, it delves into Deep Reinforcement Learning, particularly Q-Networks (DQN), which learn to make decisions by interacting with the market environment. These advanced techniques represent the forefront of predictive analytics in finance, offering new insights and capabilities beyond traditional methods.

2.3.1 Support Vector Machines

In the study conducted by Kyoung-jae Kim(2003), the research data comprised technical indicators and the daily direction of change in the Korea Composite Stock Price Index (KOSPI). The primary aim of this investigation was to predict the direction of daily price changes in the stock price index. Technical indicators served as the input variables, and 12 such indicators were initially selected based on expert insights. The central objective of this research was to anticipate the daily changes in the stock price index, which were categorized into two classes: "0" and "1." "0" denoted a scenario where the index for the next day would be lower than that of the current day, while "1" indicated that the next day's index would exceed the current day's value. The dataset spanned a total of 2,928 trading days, ranging from January 1989 to December 1998. To facilitate model development and evaluation, approximately 20% of the data (581 data points) was reserved for holdout testing, while the remaining 80% (2,347 data points) was utilized for training. The holdout dataset played a crucial role in independently evaluating the model's performance.[33]

To ensure uniformity and enhance the training process, the original data were scaled to fit within the range of [-1.0, 1.0]. This scaling technique was adopted to normalize each feature component independently, preventing any one attribute with larger values from dominating those with smaller values. This normalization process helped mitigate prediction errors. In this study, the choice of kernel functions for the Support Vector Machines (SVM) included the polynomial kernel and the Gaussian radial basis function.[33]

The study involved a comparative analysis of SVM with other methods, specifically Back-Propagation Neural Networks (BPN) and Case-Based Reasoning (CBR). The empirical outcomes clearly indicated that SVM outperformed both BPN and CBR. This superior performance can be attributed to SVM's adherence to the structural risk minimization principle, which contributes to better generalization compared to conventional techniques.[33]

In conclusion, Kyoung-jae Kim's research presents SVM as a highly promising alternative for financial time-series forecasting, with a focus on stock price index prediction. These findings and insights can be invaluable when incorporating this research into your master's thesis.

In another study with similar context, conducted by Das and Padhy (2012), the application of machine learning techniques for predicting futures prices in the Indian stock market was explored. Specifically, two prominent methods, the Back Propagation Technique (BPN) and the Support Vector Machine Technique (SVM), were employed and compared for their effectiveness in price forecasting. The study utilized real index futures data obtained from the National Stock Exchange (NSE) of India Limited, encompassing various futures contracts.[34]

To assess the prediction performance of the two techniques, several statistical metrics were employed, including the normalized mean squared error (NMSE), mean absolute error (MAE), and directional symmetry (DS). NMSE and MAE quantified the deviation between actual and predicted values, with lower values indicating a closer alignment between predicted and actual time series values. DS provided insights into the accuracy of predicted directional movements.[34]

The findings revealed that SVM consistently outperformed BPN in most cases, as evidenced by smaller NMSE and MAE values and larger DS percentages. These results suggest that SVM exhibited superior predictive capabilities compared to BPN when applied to the Indian stock market's futures price prediction, making it a noteworthy approach for financial forecasting in this context.[34]

2.3.2 Long Short-Term Memory Networks (LSTM)

Long Short-Term Memory (LSTM) models have become highly valuable tools for predicting stock market trends. They stand out because they can understand complex patterns in financial data over time. Both researchers and professionals in the finance industry are increasingly turning to LSTM models to make better predictions and improve their trading strategies. Below, we'll explore how LSTM models are used in stock market prediction, backed by important research and references.

A classification model was developed based on Long Short-Term Memory (LSTM) networks, following the approach outlined in the work by Nelson et al. (2017). This model's primary objective is to predict the price movements of multiple Brazilian stocks, specifically whether the price of a given stock will be higher or lower than its current value within a 15-minute timeframe in the future. Notably, this LSTM model is reinitialized and retrained at the commencement of each trading day, utilizing historical price data. Subsequently, it is deployed to make predictions every 15 minutes throughout the trading day, consistently employing the same model and its associated weights until the trading day concludes.[35]

The training data consisted of both past pricing data and various technical indicators. The model introduced in the paper tends to surpass the performance of the baseline methods, with only a few isolated exceptions. The results demonstrate the model's strong predictive capabilities when compared to the various approaches currently utilized in the existing body of literature. [35]

In another study conducted by Kai Chen et al. in 2015, the research focused on modeling and predicting returns in the China stock market using Long Short-Term Memory (LSTM) networks. The historical data from the Chinese stock market were transformed into sequences spanning 30 days, incorporating 10 distinct learning features alongside 3-day earning rate labels. The model was trained using a substantial dataset comprising 900,000 sequences and subsequently tested on an additional 311,361 sequences. Notably, when compared to a random prediction method, the LSTM model exhibited a significant enhancement in the accuracy of stock return predictions, increasing it from 14.3% to 27.2%. This research underscores the potential and efficacy of LSTM models in predicting stock market behavior in the context of the often-unpredictable Chinese market. [36]

2.3.3 Deep Reinforcement Learning Algorithms: Q-Networks (DQN)

Stock market prediction using *Deep Reinforcement Learning Algorithms*, particularly *Q-Networks* (DQN), has garnered significant attention in recent years. DQN is a powerful approach that combines reinforcement learning and deep neural networks to make informed decisions in dynamic and uncertain environments, making it a promising candidate for stock market prediction.

One of the primary challenges in stock market prediction is dealing with the inherent uncertainty and volatility of financial markets. Traditional models often struggle to capture the complex patterns and non-linear relationships present in financial time series data. This is where DQN comes into play. It has the ability to learn from historical data, adapt to changing market conditions, and make sequential decisions.

In their 2019 study, Chacole and Kurhekar proposed an Algorithmic Trading (AT) system employing deep Q-Learning approach. This system operates on individual stocks, allowing a single trading action per trading day, which can be "hold" (0), "long" (1), or "short" (-1), with corresponding reward structures favoring trend-aligned actions. Trading charges, accounting for 0.1% of the trading amount for both buying and selling, were considered.[37]

The research initiates by training a deep learning model with historical data, continuously enriching the training buffer with new samples each trading day. In their comparative analysis, the authors compared their proposed model against Decision Tree and Buy-and-Hold strategies. The results consistently demonstrated the superiority of the deep Q-Learning approach, exhibiting superior metrics, including percentage Accumulated Return, Maximum Drawdown, Average Return, and the Sharpe ratio. Importantly, their model showcased reduced volatility.[37]

This research underscores the potential of deep Q-Learning methodologies in Algorithmic Trading systems, particularly in single-stock scenarios. The findings highlight the promise of this approach in improving trading strategies, yielding more favorable financial outcomes, and mitigating portfolio volatility.[37]

2.4 Forecasting Stock Market using Sentiment Analysis: Recent Studies

The rapid growth of social media has significantly impacted society, drawing interest from various sectors. This has led to a complex network of interactions, fostering a platform for dialogue and collective action, giving rise to "online individualism". Social media is driving a digital revolution, impacting every individual's life. It provides a platform for self-expression and interaction, allowing individuals to share emotions and viewpoints with others. A key aspect of this is *Sentiment Analysis*, where individuals' opinions are captured and analyzed using advanced models. In essence, the goal of Sentiment Analysis is to develop automated tools capable of extracting subjective data, such as opinions, feelings, assessments, or attitudes, from text written in natural language. [38]

Sentiment, in the context of financial markets, refers to the cumulative level of optimism or pessimism expressed by market participants, which is then mirrored in the price of an asset or market at a specific moment. When the trading price of a stock or commodity significantly deviates from its intrinsic value - a value that may not be apparent until much later - this discrepancy is often attributed to sentiment. This sentiment encapsulates the collective emotional response and other abstract factors resulting from human involvement in price determination, causing it to exceed or fall short of the perceived value.[39]

This phenomenon is a focal point of research for behavioral finance, a field that explores how human cognitive biases influence financial decision-making. It's also a fundamental concept in technical analysis, a discipline that posits that prices are a fusion of factual data and emotional responses. When these emotional responses reach an extreme level, causing prices to stray significantly from the norm, a price correction is typically imminent. This correction often results in a reversion to the mean, and sometimes even further. Therefore, it's crucial for technical analysts to identify when prices are indicative of emotional extremes.[39]

In a paper by Pagolu et al. (2016) , sentiment analysis on Twitter data aimed to explore its correlation with stock market dynamics, specifically focusing on Microsoft ($MSFT). The data collection comprised 2,500,000 tweets spanning for a period of a year, gathered through the Twitter API and filtered using relevant keywords such as #Microsoft, #Windows, and $MSFT. This dataset encompassed both Microsoft-related news and tweets concerning product releases. To prepare the tweets for analysis, a three-stage preprocessing approach was employed, involving tokenization, the removal of stop words, and the elimination of special characters using regex matching. Human annotators assessed a subset of tweets, assigning emotions as 1 for Positive, 0 for Neutral, and 2 for Negative. Machine learning models were then trained using features extracted from these annotated tweets to classify non-human annotated data. [40]

Logistic Regression achieved an accuracy of 69.01%, while LibSVM, trained with 90% of the data, achieved a higher accuracy rate of 71.82%. These findings highlighted a significant relationship between stock market trends and public sentiment on Twitter, and as dataset size increased, model performance improved, suggesting potential enhancements through expanded data incorporation in future research.[37]

Another remarkable study by Valle-Cruz et al. (2022), explored how Twitter posts during the COVID-19 pandemic impacted stock markets. They collected data from January to May 2020 and compared it with Twitter's role during the H1N1 pandemic, offering insights into social media's evolving influence on stock investing.[41]

Their analysis uncovered key findings: combining a lexicon-based approach with shifted correlation analysis improved the model's performance, revealing hidden connections. Sentiments expressed on Twitter began affecting financial indices a few days after posting. SenticNet lexicon outperformed other lexicons in identifying the polarity of the posts. Effects were observed both before and after posts, showing a nuanced connection. To summarize, the result of their study highlighted the growing impact of social media platforms in predicting the stock market. [41]

3 Data & Preprocessing Methodology

The foundation of the methodology and model in this thesis begins with gathering all the necessary data. This involves collecting data from three different sources, each serving a unique role in the research. This chapter details these various datasets, explaining the reasons for their selection and the steps taken to prepare them for analysis. It focuses on straightforward processes and the rationale behind making the data ready for use, aiming to provide a clear understanding of the data collection and preparation stages.

3.1 Data Collection

For the purposes of this thesis, NVIDIA stock has been chosen for analysis. The analysis covers the period from January 2, 2020 to September 21, 2023. Data for this study were gathered from three sources: StockTwits,,Yahoo Finance and *yfinance*[1] python library. Each of these sources is described in detail in the sections that follow.

3.1.1 StockTwits

For the sentiment analysis component of the study, the StockTwits platform was utilized. Posts related to NVIDIA, tagged as $NVDA, were collected using the platform's API. StockTwits is a unique social media platform designed specifically for investors and traders. It was launched in 2008 and has since grown into a vibrant community where participants share insights, strategies, and real-time market trends. Unlike traditional social media platforms, StockTwits is tailored to the financial market, offering a focused environment for discussing stocks, bonds, cryptocurrencies, and other investment vehicles. For each post on the StockTwits platform, the study gathered a range of attributes to ensure a comprehensive analysis. These attributes include:

1. *ID:* A unique identifier for each post.
2. *Body:* The main content or message of the post.
3. *Created_at*: The original timestamp of when the post was created.
4. *User.home_country:* The home country of the user.
5. *User.followers:* The number of followers the user has on StockTwits.
6. *Likes.total:* The total number of likes the post received.
7. *Entities.sentiment.basic:* A basic sentiment analysis of the post, if available, categorizing it as bullish or bearish.

[1] https://www.pypi.org/project/yfinance/

3.1.2 Technical Indicators

Technical indicators are essential tools for traders globally, assisting them in making informed buy, sell, or hold decisions. Numerous technical indicators exist, each serving different analytical purposes. For the purpose of this study, five of the most popular and effective indicators have been selected for integration into the model to evaluate their impact on outcomes. They were fetched from the *yfinance* python library which provides real-time and historical financial market data. The chosen indicators are briefly described below.

1. Simple Moving Average Fast (SMA Fast)
2. Relative Strength Index (RSI)
3. Stoch Relative Strength Index (Stoch RSI)
4. Moving Average Convergence Divergence (MACD)
5. Volume Weighted Average Price (VWAP)

The *Simple Moving Average* (SMA) is a widely used technical analysis tool that smooths out price data by creating a constantly updated average price. This average is computed over a specific period of days, such as 10, 20, 50, 100, or 200 days. The SMA is used to identify the direction of a trend and to smooth out price volatility. A rising SMA typically indicates an uptrend, while a falling SMA suggests a downtrend. Traders often watch for crossovers of short-term and long-term SMAs as potential trading signals. For the current study the SMA Fast indicator was used which makes it more sensitive to recent price changes and allows traders to identify shorter-term trends or entry and exit points in the market.

The *Relative Strength Index* (RSI) is a momentum oscillator that measures the speed and change of price movements. It oscillates between 0 and 100 and is typically used to identify overbought or oversold conditions in a traded asset. An RSI above 70 is considered overbought, while an RSI below 30 is seen as oversold. It can also be used to spot divergences which can indicate potential trend reversals.

The *Stochastic Relative Strength Index (Stoch RSI K)* is a variation of the regular RSI, combining the features of stochastic oscillators and the Relative Strength Index. It provides a more sensitive indicator that can signal overbought and oversold conditions more frequently.

The *MACD (Moving Average Convergence Divergence)*: is a trend-following momentum indicator that shows the relationship between two moving averages of a stock’s price - the 12-day and 26-day moving averages. The MACD is calculated by subtracting the 26-day moving average from the 12-day moving average. This result is then plotted along with a signal line (a 9-day EMA of the MACD), which is used to identify buy and sell signals.

The Volume Weighted Average Price (VWAP) is a trading benchmark that is often used by investors to determine the average price a security has traded at throughout the day, based on both volume and price. It is especially important for institutional investors who need to decide whether to buy or sell large quantities without causing a significant impact on the market price. VWAP is calculated by adding up the dollar amount traded for every transaction (price multiplied by the number of shares traded) and then dividing by the total shares traded for the day. This indicator provides a comprehensive measure of a security's price, giving more weight to periods with higher volume. Traders often use VWAP to assess market direction and filter trade execution for intraday trading, considering prices above VWAP as bullish and below VWAP as bearish.

3.1.3 Historical & Financial Data

Yahoo Finance is a comprehensive financial news and data platform, renowned for offering a wide array of financial resources, including real-time stock quotes, market data, international market information, and portfolio management tools. It provides news updates, financial reports, and original content relevant to both individual investors and financial professionals. Yahoo Finance's user-friendly interface makes it accessible for tracking personal investments and conducting market research. Additionally, it offers features such as interactive charts, historical data, and live webcasts of earnings calls.

The historical data for Nvidia stock was sourced from the Yahoo Finance website. The collected data included several attributes for the same time period, including the closing price, opening price, the day's low and high prices, trading volume, and the adjusted price. It should also be noted that the dataset contains data only for the days where the Stock Market was open. Lastly, for the purposes of this study, the main emphasis was placed on the closing prices.

3.2 Preprocessing Methodology

This section outlines the crucial preprocessing steps undertaken to prepare the data for effective analysis in stock market prediction. Initially, we focus on sentiment analysis, specifically utilizing the 'Twitter-roBERTa-base for Sentiment Analysis' model to gauge the mood and opinions expressed in StockTwits posts about NVIDIA stock. This approach assigns a polarity score, reflecting the positive or negative sentiment of the market participants. Subsequently, we delve into various technical indicators calculated on a daily basis, reflecting the market's open days. These include the Simple Moving Average (SMA), which smooths price data to identify trends; the Relative Strength Index (RSI) and Stochastic RSI, both measuring the magnitude of recent price changes to evaluate overbought or oversold conditions; the Moving Average Convergence Divergence (MACD), highlighting the relationship between two moving averages of a stock's price; and the Volume Weighted Average Price (VWAP), providing an insight into the average price a stock traded at over a trading horizon, weighted by volume. Together, these preprocessing steps are fundamental in shaping the input data, ensuring that the subsequent analysis is robust, reliable, and reflective of both market sentiment and technical dynamics.

3.2.1 Sentiment Analysis – Polarity Score

In the field of sentiment analysis, there are various models available, such as *VADER* and *TextBlob*. These models are designed to determine the sentiment polarity of a text - categorizing it as positive, negative, or neutral - based on thresholds set by the user. However, for this study, a more recent and advanced model was employed: the *RoBERTa* model, which is an iteration of the *BERT* architecture.

The *RoBERTa* model represents a significant advancement in sentiment analysis. It was introduced in the paper titled '*RoBERTa: A Robustly Optimized BERT Pretraining Approach*' [42]. Building on this foundational work, several fine-tuned models have been developed, each tailored to specific types of text analysis.

For this research, the model chosen was *'Twitter-roBERTa-base for Sentiment Analysis'*, as proposed in the paper "*TWEETEVAL: Unified Benchmark and Comparative Evaluation for Tweet Classification*" [43]. This specific iteration is a *RoBERTa*-base model that has been trained on approximately 58 million tweets and fine-tuned specifically for sentiment analysis using the *TweetEval* benchmark.

The selection of this variation model was strategic for this study. The nature of the dataset from StockTwits closely resembles that of Twitter, making this model particularly well-suited for analyzing the sentiments expressed in StockTwits posts. The model’s training on a large volume of tweet data enables it to effectively classify the sentiments in short, often informal social media texts, a characteristic shared with StockTwits posts.

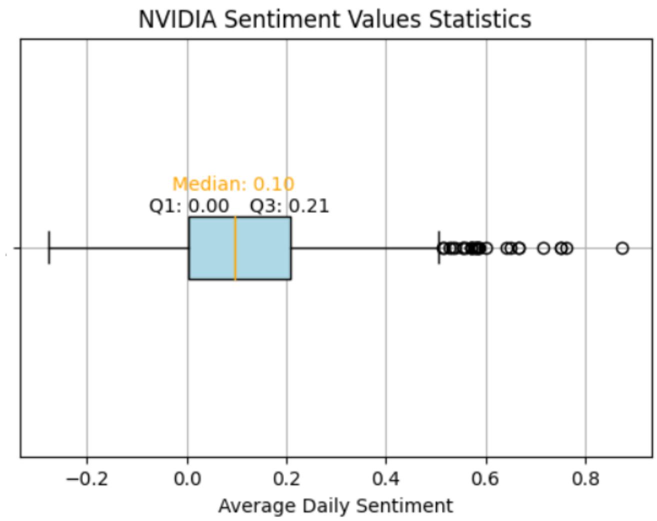

Figure 1: NVIDIA Sentiment Values Statistics

Figure 1 indicates an overall positive average sentiment for NVIDIA stock. With a scale where -1 signifies negative sentiment, 0 represents neutrality, and 1 indicates positive sentiment, the median value of 0.10 suggests that the average sentiment is slightly positive. The first quartile (Q1) represents the value below which 25% of the data falls, indicating that 25% of the data has a sentiment score equal to or less than neutral. The third quartile (Q3) at 0.21 further affirms the positive skew. The outliers, clustered to the right, reinforce the presence of particularly positive sentiment on certain days. There are no outliers or data points in the negative range, supporting the conclusion that the sentiment is generally positive. This also can be depicted from the fact that the Nvidia stock price has one of the greatest price increase in the recent years, which is something that fascinates the users.

To sum up, NVIDIA stock displays a slightly positive average sentiment which aligns with common user behavior on social media platforms like StockTwits. Users with a positive bias towards a stock are typically more vocal and engaged, often resulting in a greater volume of positive sentiment posts. This is because individuals with optimistic views may be more inclined to share their opinions and follow pages related to the stocks they favor. Consequently, sentiment analysis on such platforms may naturally skew towards the positive, as those who believe in the stock's potential are usually more motivated to contribute to the discussion.

3.2.2 Technical Indicators

This section introduces a selection of technical indicators employed in the model to analyze and predict stock market trends. These include the Simple Moving Average (SMA), which provides a smoothed understanding of price movements; the Relative Strength Index (RSI), a momentum oscillator that measures the speed and change of price movements; the Stochastic Relative Strength Index (Stoch RSI), a more refined version of RSI; the Moving Average Convergence Divergence (MACD), a tool used for identifying trend changes and potential entry/exit points; and the Volume Weighted Average Price (VWAP), which gives an average price a stock has traded at throughout the day, adjusted by its volume. Each of these indicators plays a crucial role in the analysis, offering unique insights into market behavior.

3.2.2.1 Simple Moving Average (SMA)

The *Simple Moving Average* (SMA) indicator in this study is calculated using two key parameters: the time period and the price type. The time period is set to 14 days. This means that the SMA is computed by averaging the closing prices of the stock over the last 14 days [44]. The choice of a 14-day period balances the need for capturing recent price movements while smoothing out short-term volatility.

The price parameter, referred to as 'price' in the API, is set to the 'Closing Price' of the stock. The closing price is the last price at which the stock trades during the regular trading hours on a given day. Using the closing price is a common practice in SMA calculations as it reflects the final consensus of value for that day among investors and is often considered a more accurate indicator of market sentiment compared to other prices like the opening price or high/low prices within the day.

The formula for the Simple Moving Average (SMA) is:

$$\text{SMA} = \frac{P_1 + P_2 + \cdots + P_{14}}{14}$$

Where:

- P1 + P2 + ... + P14 are the closing prices of the stock for each of the 14 days.
- The denominator represents the time period over which the SMA is calculated.

3.2.2.2 Relative Strength Index (RSI)

The *Relative Strength Index* (RSI) is an integral part of the technical analysis in this study. RSI is a momentum oscillator that measures the velocity and magnitude of directional price movements. Its primary function is to identify overbought or oversold conditions in the trading of an asset. The index oscillates between 0 and 100, offering insights into the internal strength or weakness of a stock.[45]

In this research, the RSI is calculated using specific parameters derived from an API. The focus is on a 14-day time period, which aligns with the standard practice in financial analysis. This time frame is long enough to provide a comprehensive view of the market while remaining sensitive to recent price movements.

Furthermore, the closing price of the stock is used in the calculation. The closing price is a crucial metric in financial analyses as it signifies the final trading price of the stock for the day and is often considered a reliable indicator of market sentiment.

The formula for the Relative Strength Index (RSI) is:

$$\mathrm{RSI} = 100 - \frac{100}{1 + \mathrm{RS}}$$

Where:

$$\mathrm{RS} = \frac{\text{Average Gain over 14 periods}}{\text{Average Loss over 14 periods}}$$

The RSI values range between 0 to 100, typically with thresholds set at 70 and 30. Values above 70 indicate an overbought condition, suggesting a potential price pullback, whereas values below 30 indicate an oversold condition, hinting at a potential price bounce. The application of the RSI in this study offers valuable insights into the prevailing market conditions, aiding in the decision-making process for trading strategies.

3.2.2.3 Stochastic Relative Strength Index (Stoch RSI)

The *Stochastic Relative Strength Index* (Stoch RSI) is an oscillator that measures the level of the RSI relative to its high-low range over a set time period. It provides a more sensitive indicator that can signal earlier changes in market sentiment. Developed to increase the sensitivity and reliability of the regular

RSI indicator, the Stoch RSI applies the Stochastics formula to the RSI values, rather than price values. This makes it an indicator of an indicator. The Stoch RSI is used to identify overbought and oversold conditions as well as to recognize bullish and bearish divergences. [46]

The formula for the Stoch RSI is as follows:

$$\text{Stoch RSI} = \frac{\text{RSI} - \text{Lowest Low RSI}}{\text{Highest High RSI} - \text{Lowest Low RSI}}$$

Where:

- *RSI* is the current level of the RSI.
- *Lowest Low RSI* is the lowest RSI reading for the past 14 periods.
- *Highest High RSI* is the highest RSI reading for the past 14 periods.

The Stoch RSI produces values between 0 and 1 (or 0 and 100 if scaled). A Stoch RSI reading above 0.8 is typically considered overbought, while a reading below 0.2 is considered oversold. These thresholds can be adjusted to better fit the security or analytical requirements of the trader.

3.2.2.4 Moving Average Convergence Divergence (MACD)

The *Moving Average Convergence Divergence* (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price [47]. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. The result of this calculation is the MACD line. A nine-day EMA of the MACD called the "signal line," is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals. Traders may buy the security when the MACD crosses above its signal line and sell - or short - the security when the MACD crosses below the signal line.

The formula for MACD is:

$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}$$

And the Signal Line is:

$$\text{Signal Line} = \text{EMA}_9 \text{ of MACD}$$

Where:

- *EMA12* is the 12-period Exponential Moving Average
- *EMA26* is the 26-period Exponential Moving Average
- EMA9 is the 9-period Exponential Moving Average of the MACD

3.2.2.5 Volume Weighted Average Price (VWAP)

The *Volume Weighted Average Price* (VWAP) is a trading benchmark used by traders that gives the average price a security has traded at throughout the day, based on both volume and price. It is a measure of the average price at which a stock is traded over the trading horizon. [48] VWAP is often used as a trading benchmark by investors who aim to be as passive as possible in their execution.

VWAP is calculated by adding up the dollars traded for every transaction (price multiplied by the number of shares traded) and then dividing by the total shares traded for the day. The theory is that if the price of a buy trade is lower than the VWAP, it is a good trade; the opposite is true for a sell trade.

The formula for VWAP is:

$$\text{VWAP} = \frac{\sum(\text{Price} \times \text{Volume})}{\sum \text{Volume}}$$

Where:

- *Price* is the price of each trade.
- *Volume* is the number of shares traded per each trade.
- The summation runs over all trades in the day.

VWAP serves as a reference point for the prices individual investors receive. For example, buying below the VWAP value means a relatively good price, and similarly, selling above the VWAP means a relatively good price. Due to its cumulative nature, VWAP is best suited for intraday analysis rather than over multiple days.

3.2.3 Yahoo Finance

The dataset obtained from Yahoo Finance outlines daily activities, specifically during stock market days. Notably, it includes data exclusively for days when the market is active, omitting weekends and holidays.

The primary focus of the analysis revolves around the key attribute of closing prices. A challenge arises in dealing with missing data during market closures. Two common approaches exist. One involves employing *Linear Interpolation*, estimating values for non-trading days based on prices from the preceding Friday and the subsequent Monday. Linear interpolation is a technique used for estimating values within a one-dimensional dataset. It predicts the value at a given point based on the values of the nearest points in the sequence. This method essentially uses the two points closest to the target point to form a straight line and then determines the value at the desired position along this line [49]. However, an alternative common approach is preferred in this study.

In this case, the chosen methodology involves excluding non-trading dates from the dataset entirely. This intentional omission aims to maintain a continuous and uninterrupted model environment, without disruptions caused by weekends and holidays. This approach is also discussed in the following chapters, when structuring the reinforcement model environment.

It's important to note that this practice extends to the sentiment methodology of StockTwits posts. Non-trading dates are also disregarded in the sentiment analysis preprocessing step. This ensures that StockTwits posts related to these days do not impact sentiment analysis and are not considered. This approach aims to preserve the integrity of the sentiment analysis and align with the trading dates.

4 Constructing a Reinforcement Learning Model

This chapter outlines the construction of the Reinforcement Learning model, illustrating how an agent is programmed to make decisions within a specific environment. The environment in which the agent operates, the range of actions available, and the reward system guiding its learning are discussed. Further, the implementation of the model is detailed, encompassing Experience Replay Memory for enhanced learning, a Step Decaying Learning Rate for gradual adaptation, and a Decaying Epsilon-Greedy Strategy for balanced decision-making. These elements collectively contribute to a sophisticated model designed to predict stock market trends, and are described in the following chapters.

4.1 Agent Description

In this thesis, the focus is on developing a stock market trading agent for day-to-day trading using the Double Deep Q-Network (DDQN) methodology. This advanced machine learning technique was selected over the traditional Deep Q-Network (DQN) due to its effectiveness in mitigating overestimation bias, a common issue in DQN models. DDQN achieves this by separating the action selection process from action evaluation, leading to more stable and accurate estimations of action values.

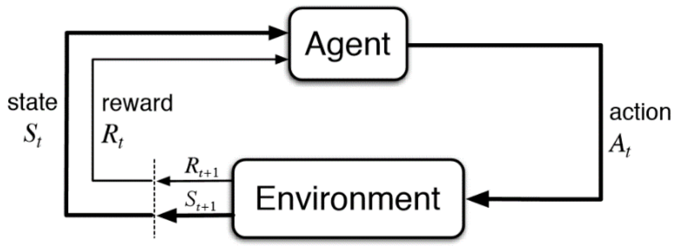

Figure 2: Schematic Representation of Q-Network Process

The DDQN agent operates with two primary actions: 'BUY' (action 0) and 'SELL' (action 1), tailored to day-to-day trading scenarios. Each trading day, the agent is required to make a strategic decision based on the market conditions. The 'BUY' action is chosen with the anticipation that the stock's value will increase, thereby allowing the agent to benefit from the subsequent rise in stock price. Conversely, the 'SELL' action is executed when the agent predicts a decline in the stock's value, aiming to avoid potential losses or secure profits from market volatility.

The reward system in this model is uniquely structured to encourage effective trading decisions. For the 'BUY' action, the reward is calculated as the difference between the stock price on the next day and the current day. A positive reward indicates a rise in the stock price following the purchase, thus validating the decision to 'BUY.' On the other hand, the reward for the 'SELL' action is computed as the difference between the current day's stock price and the next day's price. In this case, a positive reward signifies a decrease in stock price after selling, affirming the correctness of the 'SELL' action. This reward mechanism aligns with the fundamental trading principle of buying low and selling high.

The architecture of DDQN consists of two neural networks: the *policy network* and the *target network*. The policy network is responsible for making trading decisions and is continuously updated through training and interaction with the market. The target network, mirroring the policy network's architecture, provides stability during the training phase. It generates target Q-values for comparison against the policy network's predictions. The target network's parameters are periodically updated with those from the policy network, thus reducing training volatility.

Key to the agent's learning process is the *Experience* function. This function employs an *Experience Replay* mechanism, where experiences (comprising state, action, reward, and subsequent state) are stored and later sampled randomly. This technique enhances learning efficiency by breaking correlations between consecutive samples and ensuring a broad range of experiences for training.

The agent's focus on day-to-day trading is crucial, as it reflects an active trading environment where decisions are based on daily market shifts. The agent evaluates the market state each day to determine whether a 'BUY' or 'SELL' action would be most beneficial. This approach simulates a realistic trading scenario, emphasizing the importance of daily market analysis in decision-making.

To sum up, thesis introduces a DDQN-based agent designed for day-to-day stock trading. By using a relatively simple reward system and the DDQN architecture, the agent is trained to emulate real-world trading strategies, aiming to maximize profits through strategic daily decisions. This research offers insights into the application of advanced reinforcement learning techniques in the intricate world of stock market trading.

4.1.1 Environment

In Reinforcement Learning (RL), the environment is a critical component that defines the context in which an agent operates. It represents the external conditions and parameters within which the agent must make decisions. In the case of Double Deep Q-Networks (DDQN), the environment plays a pivotal role in shaping the learning and decision-making process of the agent.

The environment in a DDQN setup encapsulates the state space, action space, and reward mechanics. It presents the agent with a specific state at each step, to which the agent responds by selecting an action. The environment then evaluates this action and transitions to a new state, providing feedback in the form of rewards or penalties. This continuous interaction facilitates the agent's learning, enabling it to understand and adapt to the dynamics of the environment.

In the case study, the DDQN agent operates within three distinct environments, each progressively incorporating more market data:

1. **Closing Price Environment:** This is the most fundamental environment, where the agent's decisions are solely based on the closing price of the stocks for that day. It forms the baseline for the agent's ability to interpret basic price movements, focusing solely on the day-to-day fluctuations in stock value.

2. **Technical Indicators with Closing Price Environment:** This environment enriches the closing price data with several technical indicators: Simple Moving Average (SMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Stochastic Relative Strength Index (Stoch RSI), and Volume Weighted Average Price (VWAP). Including these technical indicators alongside the closing price gives the agent a multi-dimensional view of the market, encompassing trend analysis, momentum, and volume dynamics. This setup aims to simulate a more realistic trading scenario where decisions are influenced by a blend of price trends and technical analysis.
3. **Technical Indicators, Sentiment, and Closing Price Environment:** The most complex environment incorporates not just the closing price and technical indicators, but also market sentiment analysis. The sentiment analysis, derived from *StockTwits* social media platform, introduces an additional layer that captures market mood and investor perceptions. This combination of quantitative (closing price and technical indicators) and qualitative (sentiment) data presents a comprehensive market view for the agent, challenging it to factor in both hard data and softer, sentiment-driven market trends in its decision-making.

In all three environments, RobustScaler is used for data normalization. This scaler is particularly effective for financial time-series data, such as stock prices, which are prone to volatility and outliers. RobustScaler is adept at handling such data, maintaining the integrity of the dataset while normalizing it for effective model training. Its resilience to outliers ensures that sudden, atypical market movements do not disproportionately skew the agent's learning process. Moreover, it accommodates data with upward or downward trends, preserving the relative positioning and importance of data points during the normalization process.

Through these environments, the agent is exposed to varying degrees of market complexity, from basic price movement analysis to integrating technical, sentiment, and price data. This graded approach allows the agent to build its decision-making capabilities incrementally, adapting to increasingly sophisticated market conditions. The use of *RobustScaler* ensures a consistent and reliable input scale across all environments, enabling the agent to learn efficiently without bias towards certain features. This structured approach to environment design equips the DDQN agent to handle diverse trading scenarios, reflecting the intricate and multifaceted nature of real-world stock trading.

4.1.2 Action Space

The action space in a reinforcement learning model like the one used in this experiment plays a crucial role in defining the agent's capabilities and strategies. It refers to the range of possible actions an agent can take in a given task. Generally, action spaces fall into three categories: discrete, where actions are distinct and separate; continuous, where actions can vary in a range and are not fixed; and hybrid, a combination of both discrete and continuous elements. While pure discrete action spaces are rare in real-world scenarios, continuous and hybrid action spaces are more common and reflect the complex nature of real-world tasks [50]. In this specific case study, the action space was deliberately designed to be simple yet effective, focusing on two primary actions: 'BUY' (0) and 'SELL' (1). This binary action space aligns with the primary objective of the experiment – to determine whether the agent can effectively predict upward or downward movements in stock's price on a day-to-day basis.

The primary purpose of this experiment was not to simulate a fully realistic trading scenario. Instead, it is focused on assessing the agent's ability to make profitable decisions based on daily stock price movements. The experiment aimed to explore if the agent could learn to identify opportunities for profit by predicting whether the stock price would increase (prompting a Buy action) or decrease (prompting a Sell action) the following day.

4.1.3 Reward Function

In reinforcement learning, a reward functions as a critical feedback mechanism, providing a signal that indicates the success or effectiveness of an agent's action in its environment. It essentially acts as a guiding light, helping the agent understand which behaviors lead to positive outcomes and which do not. By receiving and analyzing these rewards, the agent gradually learns to navigate its environment more effectively, refining its strategy to maximize the cumulative reward over time. This concept is central to reinforcement learning, as it directly influences the agent's decision-making process and its ability to develop optimal policies for a wide range of tasks [51]. In this study, the reward function is designed to be straightforward yet effective, focusing on the immediate financial impact of the agent's actions based on real, unnormalized financial figures. This design choice ensures that the rewards accurately reflect the real-world implications of trading decisions, providing a clear and direct incentive for the agent to learn effective trading strategies. The reward structure is the following:

- **SELL Action**: The reward for selling is calculated as the difference between the current day's closing price and the next day's closing price of the stock. A positive reward implies that selling was a profitable decision (the price decreased the next day), while a negative reward indicates a loss (the price increased the next day).
- **BUY Action**: Conversely, the reward for buying is calculated as the difference between the next day's closing price and the current day's closing price. A positive reward here indicates a gain (the price increased the next day), and a negative reward reflects a loss (the price decreased the next day).

By calculating the rewards on actual stock prices rather than normalized values, the model offers a more realistic representation of trading outcomes. The rewards directly correspond to the financial gain or loss that would result from each trading decision, providing the agent with clear and meaningful feedback on its actions.

The simplicity of the reward function aligns with the primary objective of this study: to understand the value added by different layers of information and how they influence the agent's decision-making process. While more complex reward structures, such as managing a portfolio or incorporating a 'Hold' action, could provide a more comprehensive trading simulation, they could also introduce additional variables that might obscure the specific impact of each informational layer.

The study deliberately sets aside these complexities in favor of a clear, focused investigation into the predictive power of different data types. By limiting the action space to 'Buy' and 'Sell' and tying rewards directly to price changes, the study maintains a tight focus on the core question: how each piece of information —closing prices, technical indicators, sentiment analysis— affect the agent's ability to predict short-term stock movements.

Ultimately, this approach highlights the potential of reinforcement learning in financial markets, not only in terms of profit generation but also as a tool for understanding the intricate dynamics of stock price movements. The insights gained from this study can inform future models and strategies, paving the way for more sophisticated and nuanced approaches to algorithmic trading.

4.2 Model Implementation Details

This section dives into the specifics of implementing τηε Reinforcement Learning model. It begins with Experience Replay Memory, explaining how past experiences are utilized to enhance learning. Next, it discusses the Step Decaying Learning Rate, detailing its role in adjusting the learning process over time. Lastly, it introduces the Decaying Epsilon-Greedy Strategy, describing its function in balancing exploration and exploitation to improve decision-making. Each subsection provides insights into these critical components, demonstrating how they contribute to the model's overall performance.

4.2.1 Experience Replay Memory

Experience Replay Memory in Deep Q-Networks (DQN) is a fundamental technique in reinforcement learning, particularly in Deep Reinforcement Learning (Deep RL). This concept was popularized by the seminal paper "*Playing Atari with Deep Reinforcement Learning*" by Mnih et al. (2013) [52]. Experience

Replay is a method to store and reuse past experiences, i.e., transitions, which are in the form of (state, action, reward, next state) tuples, in the learning process.

The primary motivation behind the use of experience replay is to break the correlation between consecutive learning samples, which can significantly improve the learning stability and efficiency in Deep RL algorithms, especially when dealing with high-dimensional input spaces like video frames from Atari games (Mnih et al., 2013) [52]. By storing experiences in a replay buffer and randomly sampling from this buffer to perform learning updates, experience replay allows for more diverse and uncorrelated training batches. This random sampling process also ensures that rare but important experiences have a chance to influence the learning process multiple times, preventing the network from forgetting previously learned experiences too quickly (Lin, 1992) [53].

Experience replay in DQN works by maintaining a replay buffer (or memory), where experiences are stored as they are encountered. When the learning algorithm needs to update the network, it samples a mini-batch of experiences from this buffer.

In this case study, the Experience Replay Memory was a key component as the dataset itself was relatively small. With Experience Replay Memory, the agent has the potential to make the most out of it and utilize every bit of information hidden in the dataset. The capacity of this memory was set to 100,000 steps. This substantial capacity meant that the agent could store and access a vast array of experiences, encompassing the past 100,000 steps. Such a storage mechanism enabled the agent to retain a broad spectrum of state-action-reward-next state transitions. It allowed the agent to revisit previous experiences, enhancing its learning and decision-making processes. By leveraging these stored experiences, the agent could effectively learn from a diverse set of scenarios, ensuring a more robust and well-rounded learning.

4.2.2 Step Decaying Learning Rate

The decaying learning rate is crucial in training modern neural networks, starting high for rapid progress and escaping local minima, then decreasing to stabilize convergence. Recent insights suggest its effectiveness also lies in preventing memorization of noise and enabling the learning of complex patterns, challenging traditional beliefs [54]. The implementation of a step decaying learning rate in the DDQN model is another strategic choice aimed at enhancing the learning process. Unlike decaying alearning rate, a step decaying learning rate reduces over time at specific intervals. This method offers several benefits:

- **Efficient Convergence**: Initially, a higher learning rate is used to allow for quick convergence towards a good solution. Over time, as the learning progresses, the learning rate is reduced to allow for finer adjustments, leading to a more stable and accurate model.

- **Adaptability**: The step decaying learning rate adapts the learning process based on the agent's performance. In the initial phases, where learning is rapid, the larger steps help in faster convergence. As learning stabilizes, smaller steps help in fine-tuning the model.
- **Prevention of Oscillations**: Towards the later stages of training, a lower learning rate prevents oscillations around the optimal solution, ensuring more precise and stable learning outcomes.

The step decaying learning rate strikes a balance between exploration and exploitation, adapting the learning pace in response to the agent's evolving understanding of the environment.

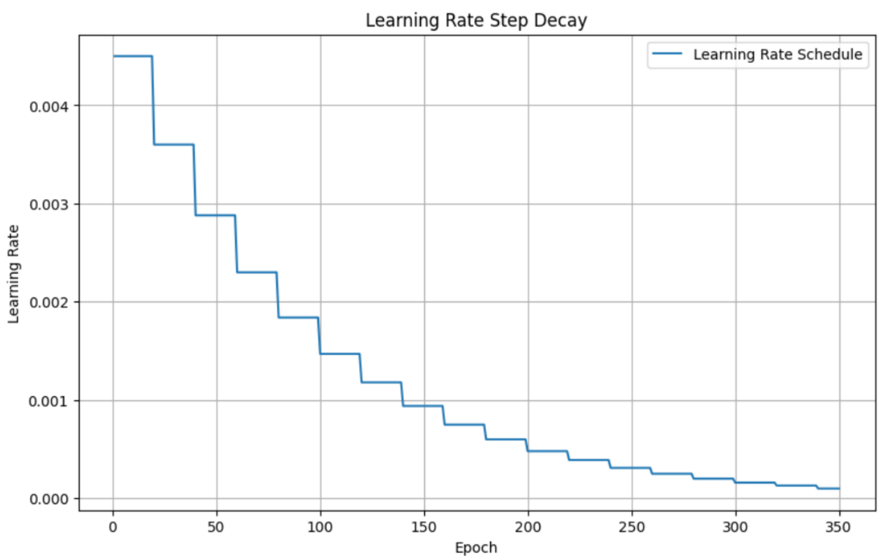

Figure 3: Learning Rate Step Decay

Figure 3 illustrates the Step Learning Rate Decay function used in the models. This function starts with an initial learning rate of 0.0045. The key characteristic of this approach is the application of a decay factor, here set at 0.8, at regular intervals - every 20 epochs as defined by the *step size*. Unlike continuous decay strategies, the learning rate remains constant for each step interval, then multiplicatively decreases by the decay factor at the end of each step. This pattern creates a staircase-like graph, where the learning rate steps down at regular intervals over 300 epochs. Such a method is beneficial for stabilizing training and allowing the model to adapt gradually to the changing learning rate, ensuring efficient and effective learning.

4.2.3 Decaying Epsilon-Greedy Strategy

The *Epsilon-Greedy* strategy in reinforcement learning, notably used in Deep Q-Networks (DQN), is designed to maintain a balance between exploring new actions and exploiting known ones. In this approach, the agent randomly selects an action with a probability determined by the value of *epsilon* (ε), and chooses the best-known action with a probability of $1 - \varepsilon$. Initially, a higher value of ε encourages the agent to explore various actions, aiding in the collection of diverse information about the environment. As discussed below, ε is usually reduced to focus more on exploiting the gathered knowledge for making optimal decisions. This simple yet effective method helps in the agent's learning by ensuring a proper balance between exploring unknown strategies and utilizing learned experiences. [55]

The *decaying epsilon-greedy strategy* in the case study DDQN model is pivotal for balancing exploration and exploitation. In the beginning, a higher epsilon value encourages exploration, allowing the agent to sample various actions and learn about the environment. Over time, as the agent gains more knowledge, the epsilon value decays, and the model shifts towards exploitation, relying more on its learned strategies.

- **Balanced Exploration and Exploitation**: This strategy ensures that the agent does not become overly conservative or risky in its actions. It maintains a healthy balance, learning from new experiences while capitalizing on existing knowledge.
- **Adaptive Learning**: The decaying aspect of the epsilon-greedy strategy ensures that the agent's behavior adapts as it learns. The rate of decay can be tuned to align with the learning progress, ensuring that the shift from exploration to exploitation is neither too rapid nor too slow.
- **Improved Decision Making Over Time**: As epsilon decays, the agent relies more on its learned Q-values to make decisions, leading to more informed and optimal choices based on its accumulated experiences.

The decaying epsilon-greedy strategy is a crucial element of the DDQN model, enabling the agent to effectively navigate the balance between exploring new strategies and exploiting known rewards, ultimately leading to a more robust and effective learning process. Figure 4 depicts the approach followed:

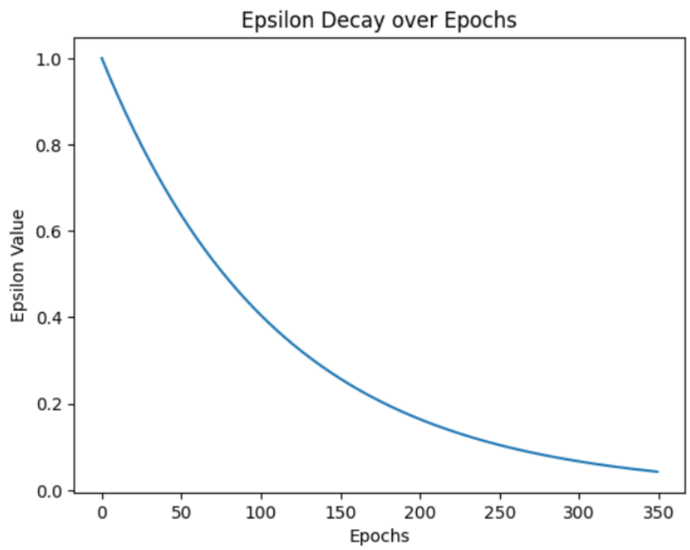

Figure 4: Epsilon Decay over Epochs

It illustrates the progression of epsilon decay over 350 epochs, a common strategy in reinforcement learning to balance exploration and exploitation. Initially set at 1.0, epsilon represents the probability of selecting a random action, fostering exploration. As training progresses, epsilon gradually decreases, following a decay rate of 0.991 per epoch with a minimum epsilon of 0.05. This decay allows for a gradual shift from exploration to exploitation, where the model increasingly relies on learned strategies rather than random choices. The plot shows a consistent, exponential decrease in epsilon.

5 Case Study: Nvidia Stock

The analysis of a Double Deep Q-Network (DDQN) agent’s performance in the stock market prediction domain requires a detailed exploration of its behavior across various training phases and environment setups. This chapter dives into the model's performance during the training and test phase for three distinct environment states, each adding a layer of complexity and information.

The first environment focuses solely on the closing price of the stock, offering a fundamental perspective on the market's movement. The second environment enhances the model's input by including technical indicators, which are vital tools in analyzing market trends and forecasting future price movements. These indicators provide the agent with insights into aspects like momentum, trends, and volatility, equipping it with a more nuanced understanding of the market dynamics. The final environment incorporates sentiment analysis, adding a layer that captures the market's psychological and subjective aspects. This inclusion aims to reflect how public sentiment on social media can influence stock prices.

For each environment in the training phase, the agent is learning to optimize its strategy based on the given state representation. The training phase is crucial, as it is where the agent develops and refines its ability to predict and capitalize on stock price movements. The performance in this phase is evaluated based on the total profits generated by the agent in every episode.

During the training phase of the Double Deep Q-Network (DDQN) model, a total of 890 active trading days were utilized, a period which accounts for days when the market was operational and excludes weekends, public holidays, and any other occasions when the market was closed. This training interval started on January 2, 2020, with the stock opening at a price of $59.97. Over the course of the training period, the stock exhibited a significant appreciation in value, culminating on July 17, 2023, when the closing price reached $464.60.

The dataset spanning this duration provided a diverse range of market scenarios, from the volatility induced by the global events of 2020 to the subsequent recovery and growth phases in the following years. This variance in market conditions likely contributed to the robustness of the training, enabling the DDQN model to learn and adapt to a multitude of trading environments.

5.1 Training Phase

The training phase is split into three different environments. Initially, an examination of the Closing Price Environment is presented. This is followed by an exploration that includes Technical Indicators, and eventually, an analysis involving both Technical Indicators and Sentiment is conducted. Within each subsection, the approach, implementation, and results are described in detail, providing insight into how various factors and data influence the effectiveness of the model.

5.1.1 Experiment Setup

This study conducts a series of experiments to evaluate the performance of the DDQN agent across three distinct environment’s states. For each environment state, three separate experiments were carried out, utilizing specific random seeds for consistency. The uniform application of these seeds across all environment’s states was crucial for ensuring the reliability and comparability of the experimental results.

Random seeds in machine learning, particularly in reinforcement learning, act as foundational points for generating sequences of random numbers. These numbers play a crucial role in various processes, such as the initialization of network weights, action selection, and sampling from the experience replay buffer.

The importance of a random seed is rooted in its ability to produce identical sequences of “random” numbers for each run, as long as the seed value remains constant.

The use of the same set of random seeds for different experiments ensures that each run is subjected to identical initial conditions and stochastic processes. This consistency is vital for comparing the performance of the agent across different environment states, reducing variability that might be attributed to random fluctuations in the learning process.

Moreover, employing fixed random seeds allows for the attribution of differences in outcomes directly to the changes in the environment states, rather than to randomness in the learning process. This aspect enhances the reliability of the findings and the conclusions drawn from the experiments.

Reinforcement learning models, such as the Double Deep Q-Network (DDQN) used in this study, are also susceptible to overfitting, especially in complex environments like stock market prediction. Overfitting occurs when a model learns the training data too well, including its noise and fluctuations, to the extent that it negatively impacts its performance on new, unseen data. This phenomenon is a critical challenge in reinforcement learning, where the balance between exploration and exploitation plays a pivotal role.

In the context of this study, the risk of overfitting was mitigated by carefully controlling the number of training episodes. The training phase was designed not only to provide sufficient exposure for the agent to learn and adapt but also to prevent it from overfitting to the specific patterns and trends of the training dataset. This approach ensures that the agent develops a strategy that is generalizable and robust, rather than one that is overly linked to the training data.

5.1.2 Experiment 1: Closing Price Environment

The initial environment in our series of experiments represents the simplest form, incorporating only the closing price of the stock. This setup serves as the most fundamental among the three distinct environments that are deployed in this study. Relying solely on the closing price, it provides a very basic level of information for the trading agent.

Given this minimalistic approach, it's clear that this environment offers limited insights. The closing price, while indicative of the stock's final market position for the day, lacks depth in terms of informing the agent about broader market dynamics. The agent, in this scenario, is excluded from essential data points that could offer a more comprehensive understanding of market trends and patterns.

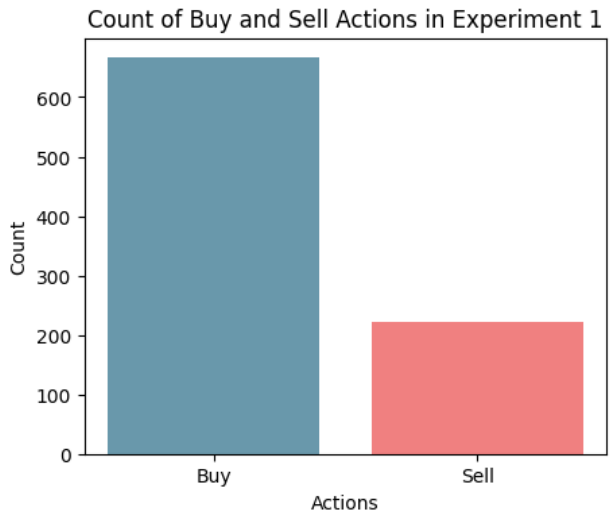

Figure 5: Experiment 1 - Buy/Sell Actions

Figure 5 shows the actions taken by the agent during the last episode. It is apparent that the agent mostly chooses the buy action, expecting higher rewards compared to sell. Without additional information, such as targeted technical indicators that could give insights on momentum or long-term trends, the agent's capacity to identify and react to market patterns is rather constrained. In such a setting, any short-term trend identified from the closing price is subject to abrupt disruptions. These disruptions can arise without any apparent reason from the agent's perspective, as it operates without the context that other critical market indicators could provide.

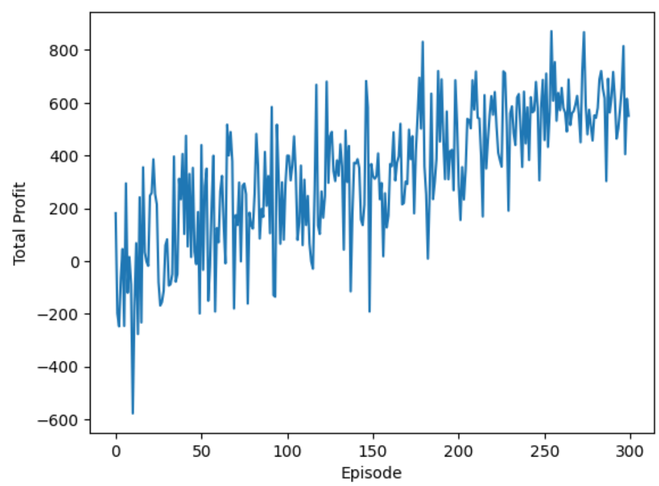

Figure 6: Experiment 1 - Evolution of Total Profits per Episode

In Figure 6, the fluctuations depicted in the progression of the episodes further underscore the agent's challenges. A significant amount of variability in total profit is observed, suggesting that the agent has yet to converge on a robust trading strategy. This instability can be a direct consequence of the environment's limited informational scope, which, lacking in-depth market indicators, leaves the agent with little to no context for making nuanced decisions.

This first environment, therefore, while serving as a starting point, is quite basic and potentially inadequate for the agent to develop a sophisticated understanding of the stock market's complexities. It underlines the necessity of enriching the trading environment with more informative attributes to enable the agent to make more informed and reliable trading decisions.

5.1.3 Experiment 2: Closing Price with Technical Indicators Environment

In the current experiment, the DDQN model's environment is enriched with technical indicators, offering a more detailed dataset for the agent. This enhancement is evident in the agent's trading actions, as shown in Figure 7, where it took 483 buy and 406 sell actions. This is a more balanced ratio compared to the initial experiment, where the agent predominantly executed buy actions. The inclusion of technical indicators has enabled the agent to better understand and respond to varying market conditions, leading to a more strategic mix of buy and sell decisions. This shift suggests an improved market analysis capability of the agent, highlighting the importance of comprehensive data inputs in refining trading strategies.

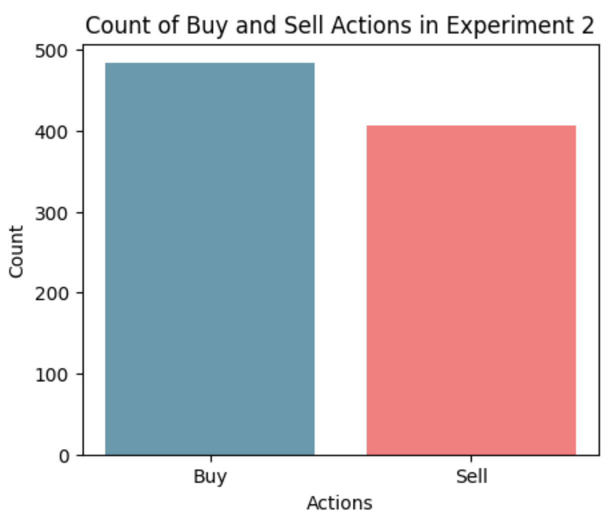

Figure 7: Experiment 2 - Buy/Sell Actions

Regarding the progress of the training, in this experiment the fluctuations per evolving episode are much more limited compared to the first experiment. This could indicate that the environment is more robust and enables the Agent to easily understand the optimal action, even in the very early episodes, meaning that the technical indicators addition is sufficient enough for the Agent to perform well.

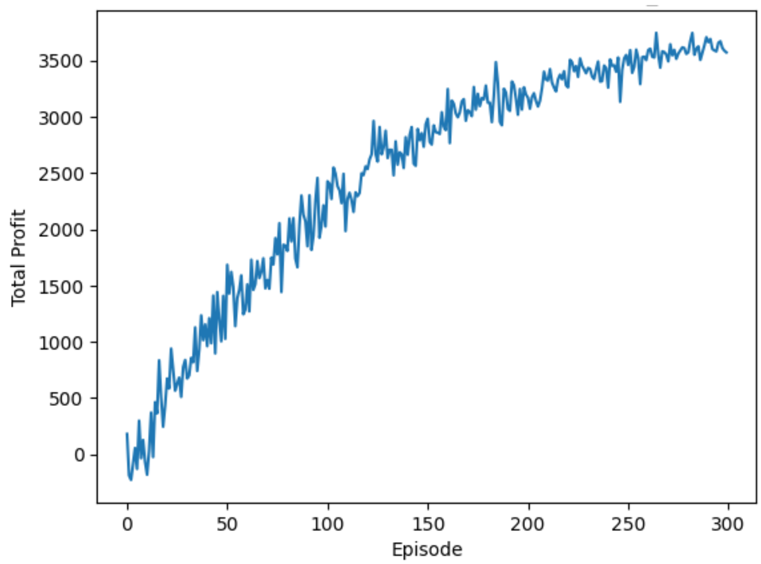

Figure 8: Experiment 2 - Evolution of Total Profits per Episode

Figure 8 depicts the evolution of total profits in training episodes. The Agent manages to identify the optimal actions much better compared to the first experiment. The total profits reach a maximum of around 3.500$ before they stabilize.

5.1.4 Experiment 3: Closing Price with Technical Indicators and Sentiment Environment

The last experiment during the training phase integrates closing prices, technical indicators, and market sentiment analysis. By combining quantitative elements (like closing prices and technical indicators) with qualitative aspects (such as sentiment), this approach offers a holistic view of the market. It challenges the agent to consider both concrete data and the more nuanced, sentiment-influenced market trends when making decisions.

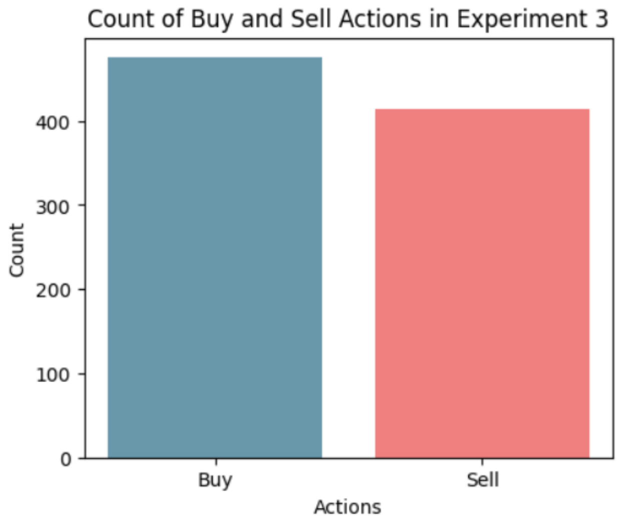

Figure 9: Experiment 3 - Buy/Sell Actions

Figure 10 illustrates the distribution of buy and sell actions taken during Experiment 3. It is evident from the chart that the 'Buy' actions, totaling 475, slightly outweigh the 'Sell' actions, which sum up to 414.

Figure 9 presents the progression of cumulative profits over a series of 300 episodes, which gained during the training phase of the Double Deep Q-Network (DDQN) model. The noticeable trend within the graph is an upward trajectory of accumulated profits, illustrating an effective learning curve for the DDQN agent. Such a trend is indicative of an increasingly proficient algorithm in executing profitable transactions within the simulated environment. The cumulative profits of this experiment seem to reach an optimal state at the last episodes, reaching more than 3.500$ after the end of the training period.

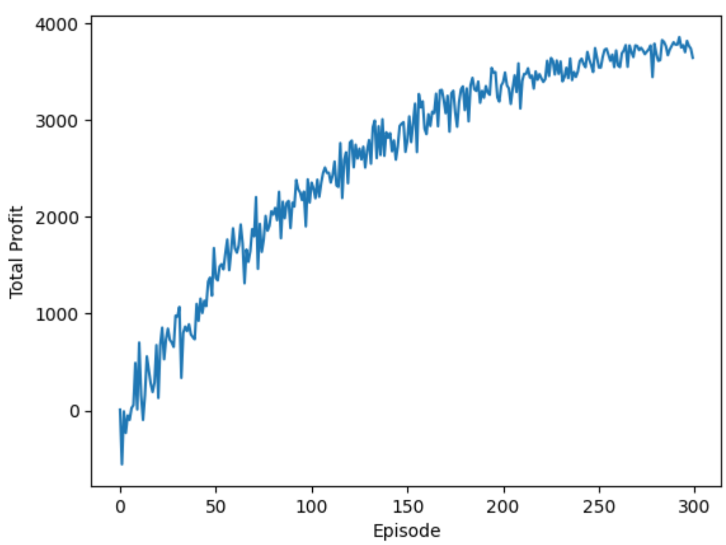

Figure 10: Experiment 3 - Evolution of Total Profits per Episode

Running the model for 300 episodes appears to be sufficient for the agent to learn and optimize its policy. The graph does not show a plateau in profit, indicating that the agent may continue to improve if given more episodes to learn from, but this approach hints a possible overfitting outcome. The total profits in the final episodes reach up to 4000$ which means that the Agent learns the optimal actions in a faster and more straightforward way, given the features in the environment.

Compared to previous experiments, τhe absence of significant fluctuations or downturns in the profit curve suggests a stable learning process. In reinforcement learning, particularly in financial environments like stock trading, stability is a desirable attribute, as it implies that the agent is not only learning but also retaining and refining his knowledge effectively. Also, the consistent increase in total profits implies that the reward function used for training the DDQN agent is well-aligned with the objective of maximizing profits.

5.2 Evaluation Phase

This section explores the Evaluation Phase, where the effectiveness of the model is rigorously assessed through a series of structured experiments. The section begins with a detailed explanation of the Evaluation Method, followed by a re-examination of the Closing Price Environment. Subsequent experiments progressively incorporate Technical Indicators and then Sentiment Analysis. Each experiment is analyzed to understand the model's performance under different conditions. Finally, key observations are compiled and presented, offering a comprehensive view of the model's capabilities and areas for potential enhancement.

5.2.1 Evaluation Method

The evaluation phase plays a pivotal role in assessing the model's efficacy. After training the DDQN agent across three distinct environments, each likely representing different market conditions or asset behaviors, the critical test of its practical applicability comes during the evaluation phase. This phase is crucial as it determines the model's ability to generalize its learned strategies to new, unseen data, a core requirement for any robust trading algorithm.

The evaluation phase, in this scenario, spans 47 trading days. These days are selected to represent a period not previously encountered by the agent during its training phase. This timeframe allows for a comprehensive assessment of the agent's performance, encompassing various market conditions that could occur over such a duration. It encompasses enough time to capture a broad range of market behaviors, including potential short-term fluctuations and longer-term trends. This period is critical to evaluate whether the DDQN model can effectively adapt and apply its learned strategies to real-world trading scenarios. The performance over these days provides a robust indicator of the model's practical utility and reliability in live trading environments, where the ability to adapt to new data is paramount.

In the realm of time series analysis, particularly in the context of financial data like stock prices, an innovative approach to normalization has gained prominence. This method, which is often referred to as *Adaptive/Dynamic Normalization* and its purpose is to address the challenges posed by non-stationary data. Traditional normalization techniques such as min-max scaling or z-score normalization are typically applied uniformly across the entire dataset. However, these methods often fall short in capturing the evolving nature of financial time series, where significant changes in scale and distribution are common.

Dynamic window-based normalization addresses this limitation by adapting the normalization parameters based on the most recent data in the training set. This ensures that the test data is normalized in a contextually relevant manner, which is particularly crucial for time series forecasting in financial markets. The dynamic nature of this method lies in its ability to account for recent trends and volatility, offering a more accurate and context-sensitive approach to normalization.

The methodology involves selecting a 'window' of recent data points from the training set. The size of this window can vary, but it typically spans a period that reflects the frequency and volatility of the data, such as the last few weeks or months for daily stock prices. The normalization parameters—such as the mean and standard deviation—are calculated based on this window. These statistics are then applied to normalize the test data. For the purposes of this study, the window of the last 30 days were taken into account.

A key advantage of this approach is its context sensitivity. By reflecting the most recent trends and volatility in the data, dynamic window-based normalization provides a more realistic and adaptable framework for data processing. This is particularly advantageous in environments where data properties are prone to rapid changes, as is often the case in the stock market.

However, the method also comes with its own set of challenges and considerations. The selection of the window size and the statistical measures used for normalization are critical decisions that can significantly impact the performance of the model. Additionally, there is a risk of overfitting to recent trends if the window chosen is too small, which might not be representative of longer-term patterns in the data. In the following chapters the results are thoughtfully presented and discussed.

5.2.2 Experiment 1: Closing Price Environment

Figure 11 illustrates the outcomes of Experiment 1 with the Agent operating under three different random seeds: 42, 75, and 93. These seeds introduce variability into the agent's training environment, testing the robustness of its trading strategy. In the first two instances (seeds 42 and 75), the agent's action distribution is remarkably consistent, with 63.8% of actions being buys and 36.2% sells. However, the third seed (93) exhibits a shift, increasing buy actions to 68.1% and decreasing sells to 31.9%. This variation suggests that while the agent's decision-making demonstrates a degree of stability, it can still be influenced by the initial conditions set by the random seed. The overall higher frequency of buy actions across all seeds could indicate a potential bias in the agent's learning algorithm or reflect the specific market conditions during the experiment. The divergence seen with seed 93 emphasizes the need for incorporating randomness in the training process to ensure the strategy's resilience and its capacity to adapt to various market scenarios.

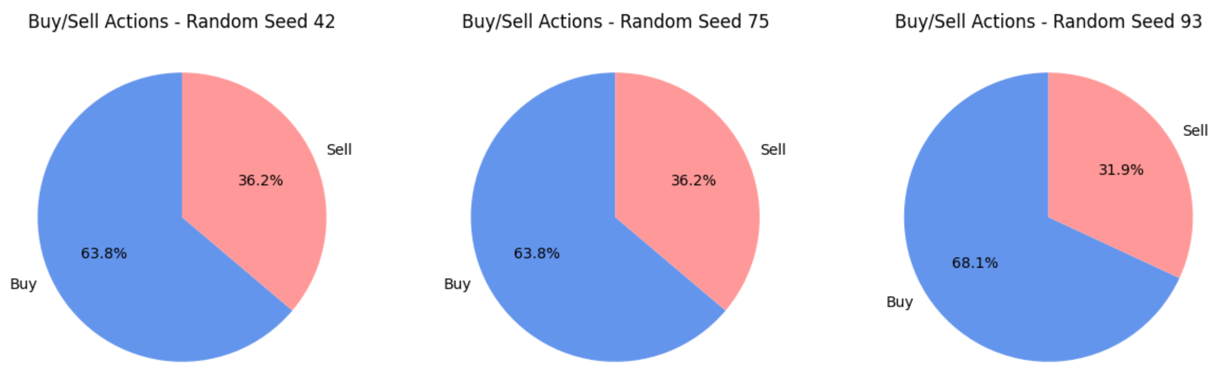

Figure 11: Experiment 1 - Buy/Sell Actions across 3 different Random Seeds

Figure 12 displays the distribution of positive and negative rewards gained by the trading agent on each test day, using the same random seeds as in the training phase. The figure reveals that the variation in rewards from day to day does not show significant differences among the three seeds. This indicates a level of consistency in how the agent's actions are rewarded regardless of the initial conditions provided by the random seeds. The rewards fluctuate similarly across all seeds, suggesting that the agent's learning and decision-making processes are stable and not overly sensitive to the randomness introduced at the beginning of the training phase.

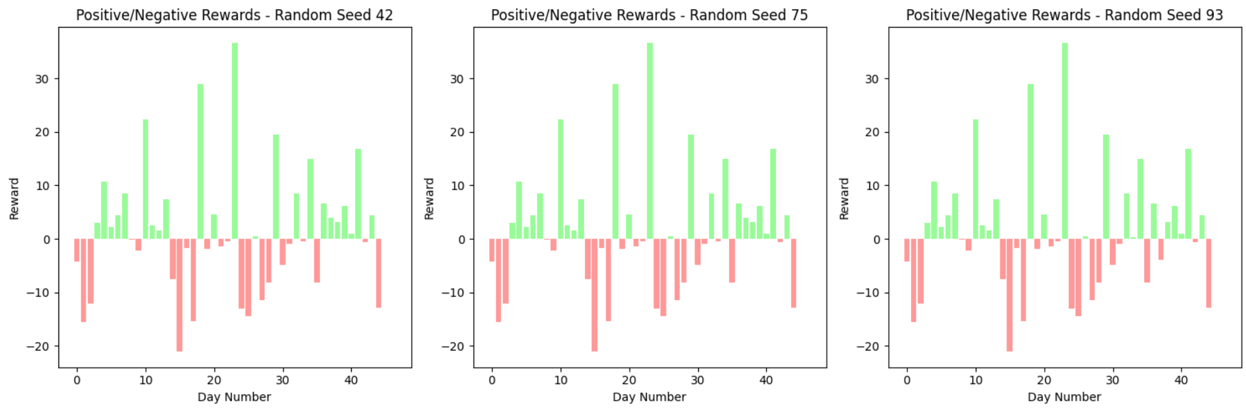

Figure 12: Experiment 1 - Daily Profits across 3 different Random Seeds

Overall, despite the simplicity of the model's environment, which is limited to the stock's closing price, the agent appears to identify many actions correctly. The distribution of rewards indicates a level of competence in the agent's decision-making ability, as it consistently receives a mix of positive and negative rewards across different seeds and days. This consistent performance reflects the agent's potential to discern and capitalize on profitable opportunities, even when operating within a constrained information set. It points to a well-constructed learning algorithm that can extract actionable patterns from a limited dataset, a promising sign for the model's application in real-world scenarios.

5.2.3 Experiment 2: Closing Price with Technical Indicators Environment

Figure 13 presents the split between Buy and Sell actions of the Agent. In Experiment 2 technical indicators complement the closing price data within the trading environment. Seed 42 shows a buy action percentage of 68.1% and a sell action percentage of 31.9%. Seed 75 presents a more balanced approach with 59.6% buy actions and 40.4% sell actions, while Seed 93 depicts 63.8% buy actions and 36.2% sell actions.

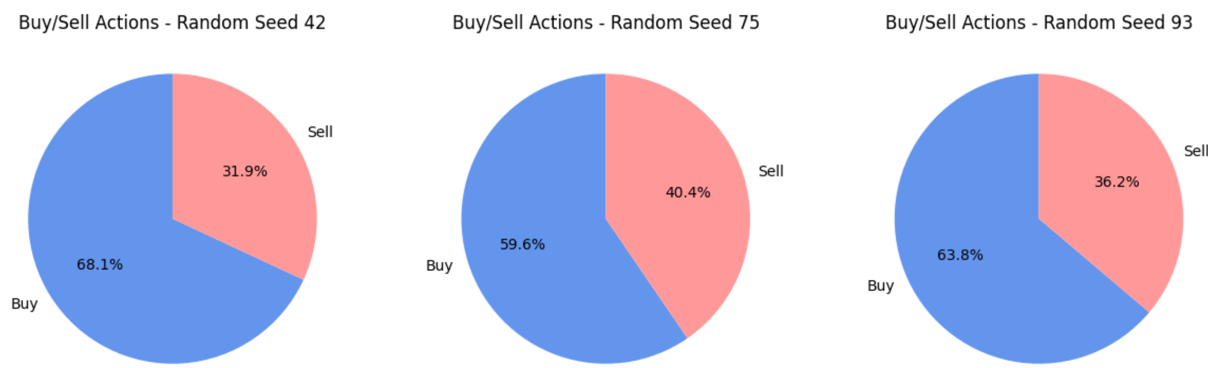

Figure 13: Experiment 2 - Daily Profits across 3 different Random Seeds

The bar charts in Figure 14 capture the daily rewards, both positive and negative, across the same set of random seeds. When comparing these results with Experiment 1, which utilized only the closing price for the trading agent's decision-making, a distinct evolution in the agent's trading behavior is evident. The introduction of technical indicators seems to have guided the agent towards a more balanced trading strategy, especially notable in Seed 75. While the agent in Experiment 1 showed a strong bias towards buying, in Experiment 2, this bias is less pronounced.

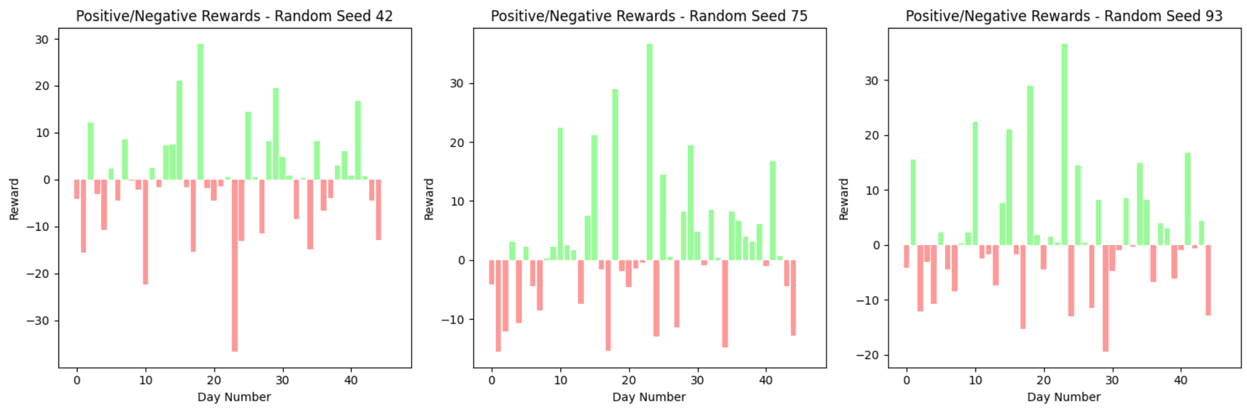

Figure 14: Experiment 2 - Daily Profits across 3 different Random Seeds

Furthermore, the rewards in Experiment 2 show a pattern of fewer extreme losses, suggesting that the additional information provided by the technical indicators may have enabled more informed and, consequently, more profitable trading decisions. The agent's ability to not only recognize short-term price trends, as was the limit in Experiment 1, but also to interpret broader market signals, has been enhanced.

The introduction of technical indicators has provided a richer informational context for the agent, enabling it to navigate the trading environment with a greater degree of sophistication. This is manifested in both the distribution of buy/sell actions and the pattern of daily rewards. The agent's capacity to earn rewards and make balanced trade decisions has improved, implying a more nuanced understanding of the market and a more robust trading strategy because of the more complex input data.

5.2.4 Experiment 3: Closing Price with Technical Indicators and Sentiment Environment

In Experiment 3, the model's environment is further enriched by incorporating sentiment analysis data from the StockTwits platform, adding a new layer of information to the closing prices and technical indicators used in Experiment 2.

Figure 15 presents the buy/sell action distribution of the trading agent with a nearly balanced split: 53.2% of actions are buys, and 46.8% are sells. This equilibrium is observed consistently across all three random seeds—42, 75, and 93—showing that the addition of sentiment analysis data has potentially enabled the agent to make decisions that are less biased towards either buying or selling compared to previous experiments.

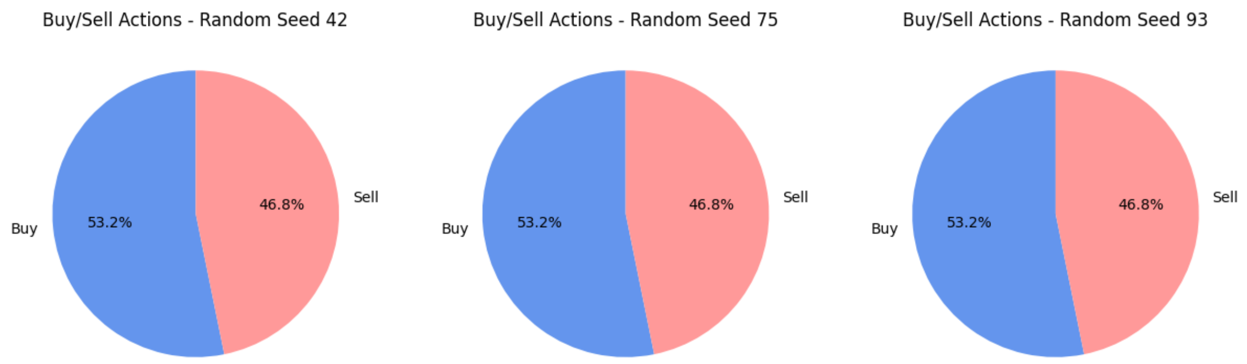

Figure 15: Experiment 3 - Buy/Sell Actions across 3 different Random Seeds

Figure 16 shows the daily positive and negative rewards for each of the random seeds. In comparison to Experiment 2, where the rewards were informed by closing prices and technical indicators, the inclusion of sentiment analysis appears to have modulated the agent's reward pattern, potentially smoothing out extreme gains or losses and providing a more nuanced understanding of market conditions that may affect trading decisions.

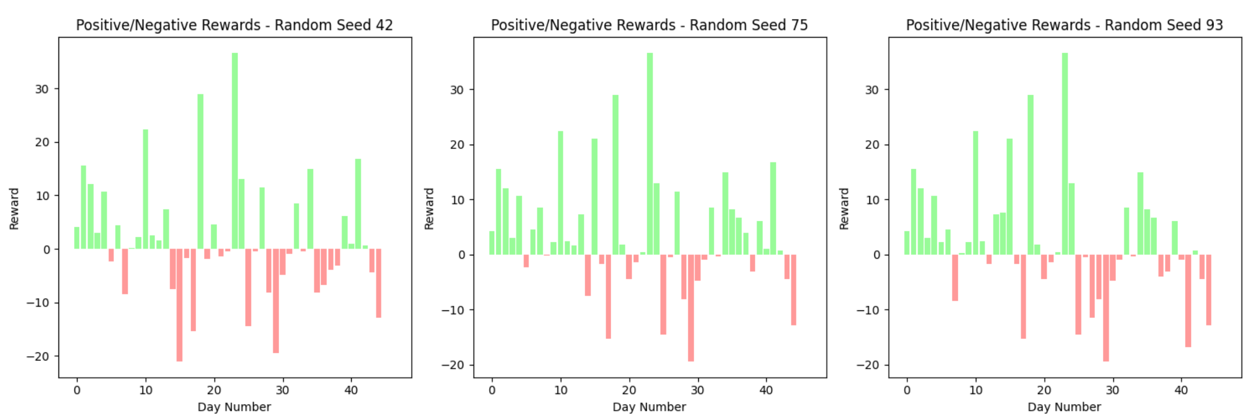

Figure 16: Experiment 3 - Daily Profits across 3 different Random Seeds

Overall, when comparing these results to Experiment 2, we can infer that the addition of sentiment analysis has had an impact on the agent's trading strategy. While Experiment 2 showed improvements in the agent's ability to balance buy and sell actions over the basic model, Experiment 3 demonstrates an even more refined approach, evidenced by the more equitable distribution of trading actions. Furthermore, the reward patterns suggest that the integration of market sentiment has given the agent an additional dimension of market insight, which enhances its decision-making process and potentially leads to more consistent performance across various market conditions.

However, it is also apparent that the added complexity of Experiment 3 has introduced variability across the three different random seeds. Although the total profits are larger compared to Experiment 2, the daily actions vary amongst random seeds, indicating a rather unstable environment.

5.2.5 Observations

Figure 11 provides a comparative analysis of the results from three different experiments conducted with the Double Deep Q-Network (DDQN) model, which is being trained and evaluated for stock market prediction capabilities. Each experiment's results are represented by a range of outcomes (minimum to maximum), depicted by the blue boxes, and the average result for each experiment, indicated by the red line.

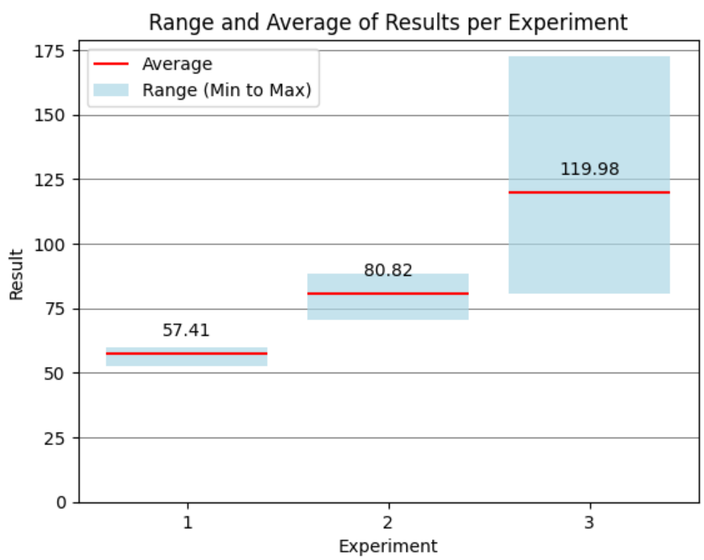

Figure 17: Comparison of Total Profits Range across Experiments & Random Seeds

Experiment 1 yields an average result of 57.41 of profits, with a relatively narrow range of outcomes. This tight range suggests that the model's performance in Experiment 1 is consistent, although with lower average results compared to the subsequent experiments. The narrow variability indicates that the environment was simple in terms of complexity, which is true as in the current experiment the environment is only the *Closing Price* of the day.

Experiment 2 shows improvement with an average result of 80.82 profits. Notably, the range of results has expanded, which could imply that the model was exposed to a more complex environment (*Closing Prices + Technical Indicators*). Still, the range of results is consistent amongst the three random seeds.

Experiment 3 stands out with a significantly higher average result of 119.98 profits, which is a substantial increment from the previous experiments. The range of outcomes is also the broadest between all experiments’ setups. This extensive variability suggests that the model was subjected to a highly dynamic environment that included multiple factors (*Closing Prices + Technical Indicators + Sentiment Analysis)*. The considerable range could also indicate that while the model achieved much higher peaks in performance, it also experienced correspondingly lower dips, possibly due to the increased complexity of the environment and the greater number of variables influencing the stock market.

The progressive increase in both the average results and the range of outcomes from Experiments 1 to 3 suggests an evolving complexity in the model's training environment and possibly an enhanced learning and adaptation by the DDQN agent. A more complex environment would presumably provide the model with a richer set of data points and scenarios to learn from, potentially enabling it to make more profitable decisions in a real-world trading context.

The growth in average results across the experiments indicates that the DDQN model is likely fine-tuning its predictive capabilities and decision-making algorithms. The increasing trend in averages and ranges suggests that the model is becoming more adept at navigating the market and possibly utilizing a richer dataset to inform its trading decisions.

However, the broad range in Experiment 3 also indicates a higher level of variability in the model's performance. This could suggest that while the DDQN model has the potential to achieve high returns, it may also be prone to significant drawdowns.

6 Conclusions and Future Work

This study aimed to understand the effect of adding layers of information on predicting NVIDIA stock's performance using a Double Deep Q-Network (DDQN) model. Initially, the model was based solely on the stock's closing price. It was then enhanced with a set of popular technical indicators commonly utilized by traders. Finally, sentiment analysis from daily posts on the social media platform StockTwits was incorporated, reflecting public opinion on the stock's future. The results clearly indicated that each added layer positively impacted the model, improving its performance and increasing the overall reward at the end of the testing period.

6.1 Conclusions

The scope of this thesis was to construct and train a Double Deep Q-Network (DDQN) model, with a specific focus on evaluating the impact of incremental information layers on the model's predictive capabilities within the stock market domain. At the heart of this exploration was the NVIDIA stock, chosen for its volatility and prominence in the market, making it an ideal candidate for assessing the efficacy of algorithmic trading strategies.

The investigation started with the simplest model environment, incorporating only the closing prices of the stock. This foundational setting served as a benchmark, establishing the baseline performance of the DDQN model without the influence of additional market variables. By starting here, the thesis could methodically assess the incremental value of each new layer of information provided. Subsequently, the environment was enriched with technical indicators to provide deeper market insights and examine their effect on the model's predictive power.

The final phase of the study introduced an interesting element: sentiment analysis. Recognizing the increasing relevance of public opinion in stock performance, the thesis attempted to quantify the sentiment of the market participants, specifically through the lens of social media commentary on the StockTwits platform. This sentiment analysis aimed to capture the mood and expectations of traders and investors, offering a daily gauge of the public opinion surrounding the NVIDIA stock.

Throughout each phase, the DDQN model's performance was documented and analyzed. The goal was to uncover the relationship between the complexity of the model's environment and its trading proficiency. By comparing the outcomes of the three distinct model environments, the thesis provides insightful revelations on the additive nature of different types of market information and their collective influence on an algorithmic trading system.

In Experiment 1, the model operates on a basic level, with closing prices as its sole input, yielding the least complex environment. As we advance to Experiment 2 and introduce technical indicators, we witness an enhancement in the agent's decision-making capabilities, as reflected by a more balanced ratio of buy and sell actions and a subsequent increase in cumulative profits. The climax of this incremental enrichment comes in Experiment 3, where the addition of sentiment analysis data injects the agent with a richer, more textured understanding of market dynamics, leading to further profitability.

However, this increase in profitability is accompanied by a proportional rise in the complexity of the model's environment. With each additional data layer, the model's interpretative and predictive capacity grows, but so does the variability in results. This variability is a natural consequence of the enhanced complexity. As more variables and more sophisticated relations are introduced into the model, the range of potential outcomes widens. The agent has to work its way through a more and more detailed mix of information, and the way all this different information comes together, can make its actions harder to predict.

The increase in cumulative profits, alongside the heightened variability, presents a tangled picture. It suggests that while the addition of information layers does result in greater profits, it also necessitates a deeper consideration of the model's environment and the robustness of the underlying trading strategies. The observed variability signals a need for caution, suggesting that the pursuit of higher profits should not overshadow the importance of understanding the nature of the increased complexity.]

6.2 Future Work

In the current thesis, the treatment of weekends and holidays—days when the stock markets are closed—was approached by excluding these periods from the analysis, thereby simplifying the dataset continuity. However, this method may overlook significant events and news that can influence stock sentiment and prices, which accumulate during these periods and can lead to abrupt changes in stock behavior once the market reopens. Future work could explore various methodologies to bridge this data gap more effectively. One such method could be to employ linear interpolation or other statistical techniques to estimate the 'missing' values, projecting the stock's trajectory based on the last available closing price and resuming with the opening price post-holiday or weekend. This could potentially allow the model to account for the momentum and market sentiment that build up during the off-market days, providing a more nuanced and continuous analysis of stock performance. Such improvements in data handling could enhance the model's predictive accuracy and offer a more robust reflection of the market's dynamics.

Additionally, the sentiment analysis in this thesis was predicated on the average sentiment of daily posts, an approach that assigns equal weight to each post regardless of the poster's influence or the post's engagement metrics. Future studies could refine this sentiment processing by introducing a weighted sentiment score that accounts for the influence of individual users. Specifically, posts from users with a significant following or recognized expertise in the domain could be given more weight, under the premise that their opinions might carry more weight in the community and potentially have a greater impact on market sentiment. Additionally, the engagement metrics of a post, such as the number of likes or reposts, could be used as proxies for its influence and relevance, thereby providing a more granular and potentially insightful sentiment measure. Such enhancements could lead to a more sophisticated sentiment analysis model that not only captures the general mood of the market but also reflects the sway of prominent voices within the investing community.

The deployment of the Double Deep Q-Network (DDQN) model within this thesis opens the door to a wide array of hyperparameter optimizations and architectural explorations for enhanced stock market prediction. Due to the newly introduced nature of reinforcement learning within the broader machine learning landscape, there lacks a standardized blueprint for fine-tuning these models to specific financial tasks. The architecture of the neural network itself, the configuration of the experience replay mechanism, and the definition of both the action space and the reward function present substantial opportunities for tailored adjustments. Allowing the agent to navigate a more complex and nuanced environment could potentially lead to the identification of subtler, yet more significant, market patterns.

Lastly, the intricacies of time series forecasting in the financial domain are vast, with numerous advanced techniques awaiting application. In this study, sliding window normalization was employed on the test data to address disparities in stock price magnitudes and trends. However, other sophisticated methods, such as '*differencing'*, could offer a more refined approach. Differencing, which involves subtracting the previous observation from the current observation, can help stabilize the mean of a time series by removing changes in the level of a time series, and thus eliminating (or reducing) trend and seasonality. Adopting such techniques in future research could enhance the model's capability to handle the inherent complexities and volatilities of financial time series data, leading to potentially more accurate forecasting models.

References

1. Nti, I.K., Adekoya, A.F. & Weyori, B.A. A systematic review of fundamental and technical analysis of stock market predictions. Artif Intell Rev 53, 3007–3057 (2020). https://doi.org/10.1007/s10462-019-09754-z

2. Strader, Troy J.; Rozycki, John J.; ROOT, THOMAS H.; and Huang, Yu-Hsiang John (2020) "Machine 2 Learning Stock Market Prediction Studies: Review and Research Directions," Journal of International Technology and Information Management: Vol. 28: Iss. 4, Article 3. DOI: https://doi.org/10.58729/1941-6679.1435

3. Boehmer, E., Fong, K., & Wu, J. (2015). International evidence on algorithmic trading.

4. Y. Xu and V. Keselj, "Stock Prediction using Deep Learning and Sentiment Analysis," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 5573-5580, doi: 10.1109/BigData47090.2019.9006342.

5. R. H. Khan, J. Miah, M. M. Rahman, M. M. Hasan and M. Mamun, "A study of forecasting stocks price by using deep Reinforcement Learning," *2023 IEEE World AI IoT Congress (AIIoT)*, Seattle, WA, USA, 2023, pp. 0250-0255, doi: 10.1109/AIIoT58121.2023.10174358.

6. Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. Machine Learning, 8, 279–292. https://doi.org/10.1007/BF00992698

7. Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence. arXiv:1509.06461v3 [cs.LG]. https://doi.org/10.48550/arXiv.1509.06461

8. Zejnullahu, F., Moser, M., & Osterrieder, J. (2022). Applications of Reinforcement Learning in Finance -- Trading with a Double Deep Q-Network. arXiv:2206.14267 [cs.LG]. https://doi.org/10.48550/arXiv.2206.14267

9. Li, X., Li, Y., Zhan, Y., & Liu, X.-Y. (2019). Optimistic Bull or Pessimistic Bear: Adaptive Deep Reinforcement Learning for Stock Portfolio Allocation. In Proceedings of the 36th International Conference on Machine Learning (ICML 2019).

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

11. Roberts, H. (1967), ‘’Statistical versus clinical prediction of the stock market”, Unpublished manuscript.

12. E. Faerber, (2008). "All about stocks," 3rd ed, Mc Grow Hill, 2008.
13. Malkiel, B. G. (2003). "The Efficient Market Hypothesis and Its Critics." The Journal of Economic Perspectives, 17(1), 59-82.
14. Bachelier, L., (1900). “THeorie de la Speculation”, Gauthier Villars, Paris
15. Fama, E. F. (1965). "The Behavior of Stock-Market Prices." The Journal of Business, 38(1), 34-105.
16. Lo, A. W., & MacKinlay, A. C. (1999). "A Non-Random Walk Down Wall Street." Princeton University Press.
17. A. G. Fontanills, and T. Gentile, "The stock Market Course,".John Wiley and Sons,Inc, 2001.
18. C. M. Thomsett, "Getting started in Fundamental Analysis," John Wiley andSons Inc, 2006.
19. Bauman, P. M., 1996. A review of Fundamental Analysis Research in Accounting. Journal of Accounting Literature.
20. Lev, B. & Ohlson, J. A., 1982. Market-based empirical research in accounting: A review, interpretation, and extension. Journal of Accounting Research.
21. Lev, B., 1989. On the usefulness of earnings and earnings research: Lessons and directions from two decades of empirical research. Journal of Accounting Research
22. Bernard, V. L., 1994. Accounting-based valuation methods, determinants of market-to-book ratios, and implications for financial statements analysis. (University of Michigan. Business School)
23. Sureshkumar KK, Elango NM (2011) An efcient approach to forecast Indian stock market price and their performance analysis. Int J Comput Appl 34(5):44–49.
24. Anbalagan T, Maheswari SU (2014) Classifcation and prediction of stock market index based on fuzzy metagraph. Procedia Comput Sci 47(C):214–221. H
25. Rajashree D, Dash PK, Bisoi R (2014) A self adaptive diferential harmony search based optimized extreme learning machine for fnancial time series prediction. Swarm Evol Comput 19:25–42.
26. Bisoi R, Dash PK (2014) A hybrid evolutionary dynamic neural network for stock market trend analysis and prediction using unscented Kalman flter. Appl Soft Comput J 19:41–56.

27. Murphy, J. J. (1999). "Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications." Penguin.

28. Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2015). "Time Series Analysis: Forecasting and Control." John Wiley & Sons.

29. Jegadeesh, N., & Titman, S. (1993). "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency." The Journal of Finance, 48(1), 65-91.

30. Moskowitz, T. J., & Grinblatt, M. (1999). "Do Industries Explain Momentum?" The Journal of Finance, 54(4), 1249-1290.

31. Chan, L. K., Jegadeesh, N., & Lakonishok, J. (1996). "Momentum Strategies." The Journal of Finance, 51(5), 1681-1713.

32. Andreassen, P., Kraus, S., 1990. Judgmental extrapolation and the salience of change. Journal of Forecasting 9, 347—372.

33. Kim, K. J. (2003). Financial time series forecasting using support vector machines. Neurocomputing, 55(1-2), 307-319.

34. Das, S. P., & Padhy, S. (2012). Support vector machines for prediction of futures prices in Indian stock market. International Journal of Computer Applications, 41(3).

35. Nelson, D. M., Pereira, A. C., & De Oliveira, R. A. (2017, May). Stock market's price movement prediction with LSTM neural networks. In 2017 International joint conference on neural networks (IJCNN) (pp. 1419-1426). Ieee.

36. K. Chen, Y. Zhou and F. Dai (2015), "A LSTM-based method for stock returns prediction: A case study of China stock market," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 2015, pp. 2823-2824, doi: 10.1109/BigData.2015.7364089.

37. Chakole, J, Kurhekar, M. (2020), “Trend following deep Q-Learning strategy for stock trading. Expert Systems.”, 37:e12514. https://doi.org/10.1111/exsy.12514

38. Mitra, G., & Xiang, Y. (2016). The Handbook of Sentiment Analysis in Finance. Albury Books.

39. Kirkpatrick II, C. D., & Dahlquist, J. R. (2010). Technical Analysis: The Complete Resource for Financial Market Technicians. FT Press.

40. Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2016, October). Sentiment analysis of Twitter data for predicting stock market movements. In 2016 international conference on signal processing, communication, power and embedded system (SCOPES) (pp. 1345-1350). IEEE.

41. Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A., & Sandoval-Almazán, R. (2022). Does twitter affect stock market decisions? financial sentiment analysis during pandemics: A comparative study of the h1n1 and the covid-19 periods. Cognitive computation, 1-16.

42. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]. https://doi.org/10.48550/arXiv.1907.11692

43. Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., & Neves, L. (2020). TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 1644–1650). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.148

44. Hayes, A. (2023, September 30). Simple Moving Average (SMA): What It Is and the Formula. Investopedia. [https://www.investopedia.com/terms/s/sma.asp]

45. Fernando, J. (2023, March 31). Relative Strength Index (RSI) Indicator Explained With Formula. Investopedia. [https://www.investopedia.com/terms/r/rsi.asp]

46. Hayes, A. (2021, August 2). Stochastic RSI -StochRSI Definition. Investopedia. [https://www.investopedia.com/terms/s/stochrsi.asp]

47. Dolan, B. (2023, December 19). MACD Indicator Explained, with Formula, Examples, and Limitations. Investopedia. [https://www.investopedia.com/terms/m/macd.asp]

48. Fernando, J. (2024, January 7). Volume-Weighted Average Price (VWAP): Definition and Calculation. Investopedia. [https://www.investopedia.com/terms/v/vwap.asp]

49. Huang, G. (2021, February). Missing data filling method based on linear interpolation and lightgbm. In Journal of Physics: Conference Series (Vol. 1754, No. 1, p. 012187). IOP Publishing.

50. Zhu, J., Wu, F., & Zhao, J. (2022). An Overview of the Action Space for Deep Reinforcement Learning. In Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence (ACAI '21) (Article 52, pp. 1–10). Association for Computing Machinery. https://doi.org/10.1145/3508546.3508598

51. Icarte, R. T., Klassen, T. Q., Valenzano, R., & McIlraith, S. A. (2022). Reward machines: Exploiting reward function structure in reinforcement learning. Journal of Artificial Intelligence Research, 73, 173-208.

52. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602 [cs.LG].

53. Lin, L. (1992). Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. Machine Learning, 8, 293–321.

54. You, K., Long, M., Wang, J., & Jordan, M. I. (2019). How does learning rate decay help modern neural networks?. arXiv preprint arXiv:1908.01878.

55. Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.