An AI Trading Bot to Increase Investment Portfolio Performance

MSc Artificial Intelligence
Practicum 2

Rodrigo Esparza Giacomelli

Student ID: X23362804

School of Computing
National College of Ireland

Supervisor: Simiscuka A

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rodrigo Esparza Giacomelli		
Student ID:	X23362804		
Programme:	MSCAI1B	**Year:**	2025……..
Module:	Practicum 2 ……………………………………….		
Supervisor:	Simiscuka A ………………………………………………		
Submission Due Date:	11 – August- 2025 ……………………………………………………………….………		
Project Title:	An AI Trading Bot to Increase Investment Portfolio Performance		
Word Count:	…………… ………………………… **5745**……………………………………….……..		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	……………Rodrigo Esparza Giacomelli………………………………………
Date:	11 August 2025………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An AI Trading Bot to Increase Investment Portfolio Performance

Rodrigo Esparza Giacomelli
23362804
National College of Ireland

Abstract

The volatility in the stock market, understood as the unpredictable variation of prices, makes it significantly harder to make accurate investment decisions. However, different AI techniques have started to be used to detect patterns in the market and provide a stronger base for making investment decisions. Still, the wide variety of ways these tools can be applied to decide the best times to buy or sell makes their practical use a big challenge. This work proposes different neural network implementations to increase portfolio investment through trading. The neural networks that will be implemented will use the DQN reinforcement learning algorithm. In these networks, different financial strategies and architectures will be tested to find which model gives the best performance with more stable and secure behaviour. After that, the best DQN model will be combined with an NLP pipeline that analyses international news and predicts its impact on the stock market, creating a final neural network that uses both quantitative and qualitative data for trading. The results will show which neural network architecture, training method, exploration parameters, and financial strategy are the best to use for trading, and how much the model can change if international news is taken into account or if only the numerical history of the stock is considered.

Keywords – Stock Market, Reinforcement Learning, DQN, Deep Learning, NLP, Volatility

Introduction

It is estimated that more than 70% of the transactions in the main financial markets are executed by automated trading systems. However, developing effective strategies that generate consistent returns is still a challenge, due to high volatility, the complexity of financial data, and the unpredictability of the markets. In recent years, deep learning algorithms like DQN have shown high potential for decision-making in sequential environments, despite the difficulty that exists in making the right decisions for proper implementation due to the wide variety of possibilities that can affect the model's performance and reliability

Some recent approaches have tried to complement numerical market data with textual information extracted from international news using NLP techniques. However, the processing and effective integration of textual data present significant challenges, such as semantic ambiguity and the lack of temporal synchronization with market movements

The aim of this research is to determine how effective the application of the Deep Q-Network (DQN) algorithm is in neural networks to identify patterns in historical stock market data and to maximize the profits obtained through automated trading strategies. Some of the research questions that guide this project are:

- RQ1: Which Deep Q-Network architecture delivers the highest portfolio returns and risk-adjusted performance in automated stock trading?
- RQ2: To what extent does integrating NLP news analysis improve a DQN trading agent's decision-making profitability compared to a purely numerical data driven approach?

To address these research questions, the following specific sets of research objectives were derived:

1. Investigate key financial aspects for making buy and sell decisions on stocks, market behaviors in relation to external factors, and main AI algorithms for trading strategies using neural networks.
2. Design different neural network architectures, as well as training and evaluation environments, considering different configurations and training parameters to analyze their impact on the model's performance.
3. Implement the designed architectures and set up the defined environments, ensuring their correct operation for training and validating the models.
4. Evaluate the final value of the portfolio, Cumulative Return, Maximum Drawdown, Sharpe Ratio, Profit Factor, Win Rate, and Final Reward of each implementation in different time periods to determine the best trading model.

The major contribution of this research is a novel trading model based on the DQN algorithm, specifically designed to operate in the stock market. After evaluating different implementations, it was shown that the selected model offers greater stability and security to maximize investment performance in any type of market behavior, during instability, when the market is going up, or when it is going down. A minor contribution of this research is the implementation of an NLP pipeline focused on the analysis of international news, evaluating the possible impact that the news could have on a specific asset. It generates parameters such as the level of volatility it could cause (low, medium, or high), the relevance of the news regarding the affected stock (on a scale from 1 to 10), and the sentiment analysis of the content.

This paper discusses different Deep Learning implementations that can be used to create a Neural Network for Trading, considering the analysis of international news through NLP. Section 2 mentions the related works that were considered. The research methodology is discussed in Section 3. Section 4 discusses the design component for the deep learning framework. Section 5 discusses the implementation of the research. Section 6 presents the results and evaluations. Section 7 shows the conclusions and discusses future works

Related Work

The best way to use artificial intelligence for stock market prediction has been sought, along with qualitative text analysis, Eid M et al. integrated empirical use cases and theoretical frameworks analyzing artificial intelligence techniques such as machine learning and natural language processing for sentiment analysis in financial news, as well as rule-based systems for structured decision-making, and conclude it that the implementation of AI reduces human errors in financial diagnostics and enables greater scalability in logistics route optimization [16]. Also, Ashraf M et al. analyzes the use of AI to improve portfolio performance, overcoming the limitations of traditional investment models such as Markowitz's Mean-Variance Model, exploring machine learning techniques such as SVM, Random Forest, LSTM, and Reinforcement Learning, showing that these models outperform traditional approaches, with Random Forest and RL standing out as the most effective [1], and S. Venkatesan et al. also proposes the use of artificial intelligence to manage investment portfolios, using the DQN algorithm to maximize returns and reduce risks. This strategy outperforms traditional methods, achieving a 12.5% ROI compared to 9.8%, and a Sharpe Ratio of 1.2 versus 0.9 when compared to static approaches [2].

Multiple studies have been conducted to determine the best implementation of artificial intelligence that can be applied to optimize performance and increase investment returns in a safe and stable way. Fahima H et al. proposes various deep reinforcement learning (DRL) models to predict stock market behavior, comparing five different models: A2C, DQN, ACKTR, PPO1, and PPO2, and optimizing financial indicators to evaluate the results across 20 different stocks from 2000 to 2004. The results show that the DQN model is the most effective in dynamic environments, achieving the best predictions in terms of investment return, particularly for MSFT and NVDA [3]

Mojitaba N et al. compares the performance of 11 different algorithms, 9 from machine learning (ML) and 2 from deep learning (DL), to predict trends in stock markets. The evaluation is based on 10 technical indicators and the normalization of these indicators to the range (-1, 1). The ML models used include Decision Tree, Random Forest, AdaBoost, XGBoost, SVC, Naïve Bayes, KNN, Logistic Regression, and ANN, while the DL models are RNN and LSTM. The results show that RNN and LSTM achieved the best performance, with F1-Scores of 86% and 90% respectively, and their performance improved with data normalization [4]

Victor C et al. compares the performance of three different Deep Learning and Reinforcement Learning models: DQN, Double DQN, and Dueling Double DQN, to determine which one performs best in stock trading. Various profit analyses, loss and reward functions, and technical indicators were tested, showing that the DQN algorithm achieved the best performance in predicting stock market behavior and delivered higher returns in trading operations [5]

Kumar A et al. proposes a market prediction model using neural networks and compares the performance of different architectures, using networks with 3, 4, and 5 hidden layers, each with 20 nodes. The results showed that the model with 3 hidden layers was the most effective in predicting market behavior, achieving a mean squared error (MSE) of 0.00356, which indicates it is an efficient tool for forecasting stock market trends [6]

Taylan K et al. proposes a Deep Reinforcement Learning approach to automate stock market trading, treating it as a partially observable Markov decision process (POMDP). Financial indicators such as RSI, SMA, EMA, %K, MACD, A/D, OBV, ROC, Williams %R, and Disparity Index were integrated, achieving positive results with a Sharpe ratio of 3.14, outperforming the benchmark of 2.4 [11]

Junhao Z et al. compares DQN with Policy Gradient, two deep reinforcement learning algorithms to predict movements in market prices, using financial indicators like opening price, closing price, high, low, daily price, and trading volume. In this comparison, the PG model shows better performance in tests but also higher volatility at the same time, while DQN remained more stable and accurate in decision-making [12]

Leandro M et al. analyzed the application of neural networks to predict stock indices in North America, Europe, and Brazil, testing different architectures with 1 to 2 hidden layers of 2 to 6 nodes each, comparing their performance against GARCH models using metrics like RMSE, R2, and POCID. The results showed that, on average, neural network implementations outperformed GARCH models with an average R2 of 0.96 versus 0.83 and a POCID of 75% versus 60%. [14]

There have also been studies that aim to predict stock market behavior by analyzing financial news, political texts, or other information through NLP to assess the impact they might have. Oruganti J et al. carries out a systematic review of NLP methods for sentiment classification in online news, comparing traditional techniques like SVM and Naive Bayes, and advanced models such as RNN, LSTM, and BERT. These methods were applied to financial and journalistic texts, showing promising results for BERT and SVM models, which outperformed the others in accuracy for understanding news, especially in financial texts. [18]

Chen C et al. investigates how to use NLP together with ML to predict stock market movements using Reddit news headlines. It includes text processing with tokenization, stopword removal, and lemmatization, sentiment analysis with VADER, and data transformation using TF-IDF, combining these with Random Forest and Logistic Regression. The models were evaluated with metrics like recall and F1-Score, showing good ability to handle small datasets and a solid probabilistic approach, proving to be an optimal solution for this type of problem [7]

Kevin T et al. explores different NLP models on news to predict stock price trends, based on the percentage change of the stock. It uses techniques like tokenization with BERT-Tiny across six different text models, including headlines, sources, dates, abstracts, authors, etc. The results show that for analyzing a news article, the best text model to use is only the headlines and company names, since adding more text can cause noise in the prediction and reduce accuracy [8]

Jiaozhou Q discusses the implementation of NLP combined with supervised and unsupervised ML algorithms to merge quantitative data analysis with qualitative news analysis, extracting information from financial news, reports, and political documents. These parameters are applied in graph neural networks to find correlations between assets and market trends. The results show that combining these two models significantly improves investment portfolio performance, highlighting the model’s ability to identify asset correlations and adapt to changing market environments [10]

In conclusion, the implementation of neural networks for stock market prediction is a viable application, since it outperforms static financial strategies by detecting patterns in the market's behavior and adapting to different conditions. Among the different applications of neural networks, the use of Deep Learning and Reinforcement Learning stood out as the most optimal method to implement these models, with the DQN algorithm showing the best performance in Trading, delivering more stable and consistent results compared to the other algorithms.

On the other hand, the use of NLP to obtain the relevance of news and perform sentiment analysis to detect the impact that a text could have on a specific stock showed promising results. By using Vader and TF-IDF to evaluate the relevance of the impact, solid and consistent results were achieved.

So, it proves to be a viable method to use both of these models to predict the stock market and support different financial strategies and AI tools in order to improve the performance of financial portfolios, generating more trust in investments, more stable growth, and better return on investment

Research Methodology

The research methodology consists of five steps namely data gathering, data pre-processing, data transformation, data modelling and conversion, evaluation and results as shown in Fig 1

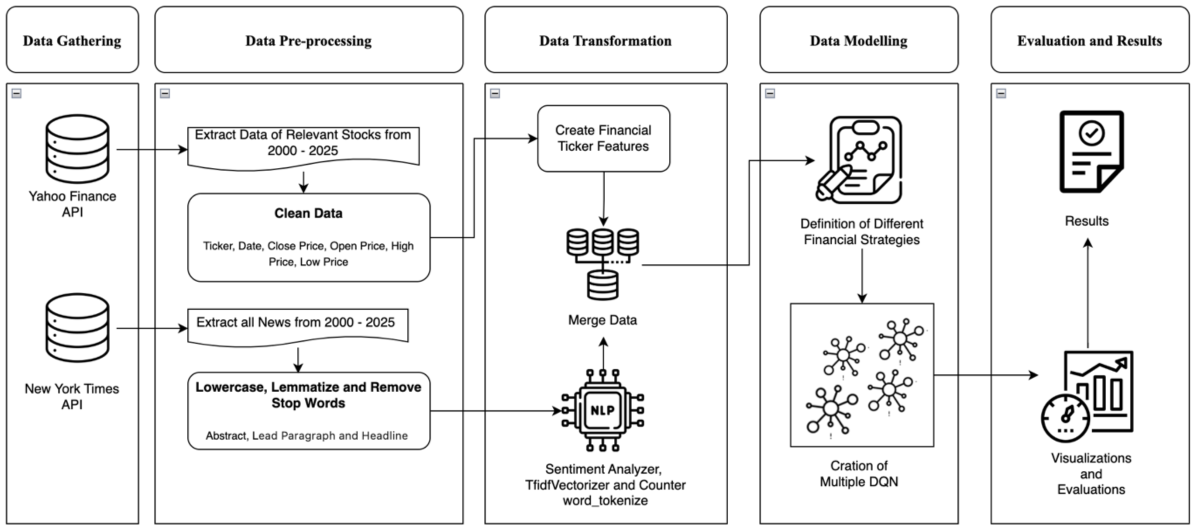

Fig. 1. Research Methodology

The first step, Data Gathering involves collection of financial information from Yahoo Finance API using the Python library yfinance, and we extracted "Date," "Open Price," "High Price," "Low Price," "Close Price," and "Volume" from the ticker 'SPY', which corresponds to the S&P 500. We selected the data range from 2000 to 2025 to have the last 25 years of information for this stock.

For the news collection, we connected to the New York Times API using the endpoint https://api.nytimes.com/svc/search/v2/articlesearch.json, which provides all the articles published by this outlet. We selected the range from 2000 to 2025 to store in a dataset more than 2 million records of information provided by the API. This dataset can also be found in the following Kaggle repository: https://www.kaggle.com/datasets/aryansingh0909/nyt-articles-21m-2000-present

The second step, Data Pre-processing involves visualizations of the financial data to observe the behavior of the collected information and then cleaned by filtering it by ticker and selecting the information relevant to the experiment, removing missing values and irrelevant columns. As for the information collected from international news, a data analysis exploration was first performed, where the date range of the news, the number of articles, the different categories, and other comparisons relevant to the experiment were examined. Then, the dataset was cleaned by selecting only the important columns such as "Abstract," "Lead Paragraph," and "Headline," and combining them into a single variable, to then apply Lowercase, Lemmatize and Remove Stop Words to leaving it ready to apply different NLP processes.

The third step, Data Transformation involves the information collected from Yahoo Finance to create different daily financial indicators relevant for applying trading strategies, such as “RSI,” “Bollinger b,” “MACD,” among others. These parameters were used as features to train the DQN models. Besides creating these features, NLP processes were applied to the collected news to obtain the sentiment of each one. Tokenization was also done to predict the relationship that the text has with the S&P 500 and the level of volatility the news could cause.

After that, the parameters obtained through NLP in international news were combined with the features generated from financial numerical data to have all the information in a single dataset ready to start creating the DQN Model.

The fourth step, Data Modelling involves different financial trading strategies currently used were analyzed to replicate their behavior in neural networks. So, the process started from the simplest strategies to the most complex ones, implementing different NN architectures, reward systems, exploration parameters, financial indicators, and actions to be taken, applying the DQN algorithm to generating 4 different neural network models with different behaviors.

The fifth step, Evaluations and Results involves evaluating the performance of each of the DQN models using Final Value of the portfolio, Cumulative Return, Maximum Drawdown, Sharpe Ratio, Profit Factor, Win Rate, and Final Reward. The five models were compared and visualized using matplotlib. The optimal DQN model was selected from the experimentation 6 and implemented in the demo.

Design Specification

The neural network architectures used to create the DQN models with different financial strategies can be seen in Fig. 2, where the different implemented strategies are shown. In Fig. 3, the architecture used for applying NLP to analyze international news is shown, and how it was combined into a new neural network that mixes both financial information and news text to perform trading. A detailed explanation can be found in Section DQN Financial Models and DQN and NLP News Model

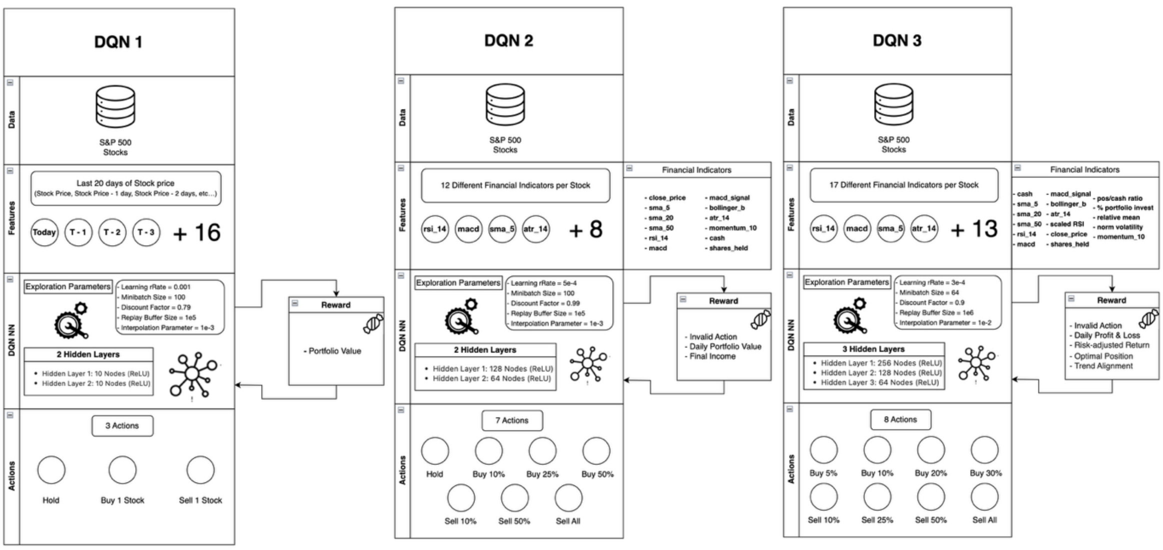

Fig. 2. DQN Architectures

DQN Financial Models

DQN 1 architecture involves a neural network focused only on financial information, so in the first layer, the input uses the last 20 stock prices (Stock Price, Stock Price - 1 day, Stock Price - 2 days, etc…), which means it has a layer of 20 nodes. Besides this, it has 2 hidden layers with 10 nodes each, as it was implemented in the work “Application of Deep Reinforcement Learning on Automated Stock Trading" by Lin Chen and Qiang Gao in 2018. The training parameters for exploration were Learning Rate = 0.00, Minibatch Size = 100, Discount Factor = 0.79, Replay Buffer Size = 1e5, Interpolation Parameter = 1e-3. The reward mechanism used was based on the daily value of the portfolio, so it received a positive reward if the total value of the portfolio was higher than the previous day, or negative if it was lower. Finally, the actions of this NN were 3: “Hold,” “Buy one stock,” and “Sell one stock,” so this model is limited to only modifying the portfolio of one stock per day

For DQN 2, the goal was to improve the performance of the neural network. So, in the first input layer, 12 typical financial indicators used in trading decision-making strategies were implemented, such as close_price, sma_5, sma_20, sma_50, rsi_14, macd, macd_signal, bollinger_b, atr_14, momentum_10, cash, and shares_held. The number of nodes in the hidden layers was also modified, making the first layer 128 nodes and the second one 64 nodes. The Learning Rate and Discount Factor were changed to 0.0001 and 0.99. Indicators were added to the reward to improve the NN’s learning experience by implementing penalties for Invalid Actions and a final reward to evaluate the simulation’s final result. Finally, more actions were added, leaving 7 possible final actions: “Hold,” “Buy 10%,” “Buy 25%,” “Buy 50%,” “Sell 10%,” “Sell 50%,” and “Sell All.”

Finally, to improve performance, the DQN 3 model was created. In this version, 5 relevant financial indicators were added to help stabilize trading decisions. These were: pos/cash ratio, % portfolio invest, relative mean, norm volatility, and momentum_10. An extra hidden layer was also added to detect more complex patterns in the stock behavior, resulting in three layers: one with 256 nodes, another with 128, and the last one with 64. The training exploration parameters were also modified to the following: Learning Rate = 3e-4, Minibatch Size = 64, Discount Factor = 0.9, Replay Buffer Size = 1e6, and Interpolation Parameter = 1e-2. A completely new reward system was created, which evaluates the NN’s behavior in 5 different aspects, assigning an importance weight to each one, giving the following: Invalid Action = 15%, Daily Profit & Loss = 40%, Risk-adjusted Return = 30%, Optimal Position = 10%, and Trend Alignment = 5%. Finally, the actions were modified to: “Buy 5%,” “Buy 10%,” “Buy 20%,” “Buy 30%,” “Sell 10%,” “Sell 25%,” “Sell 50%,” and “Sell All.”

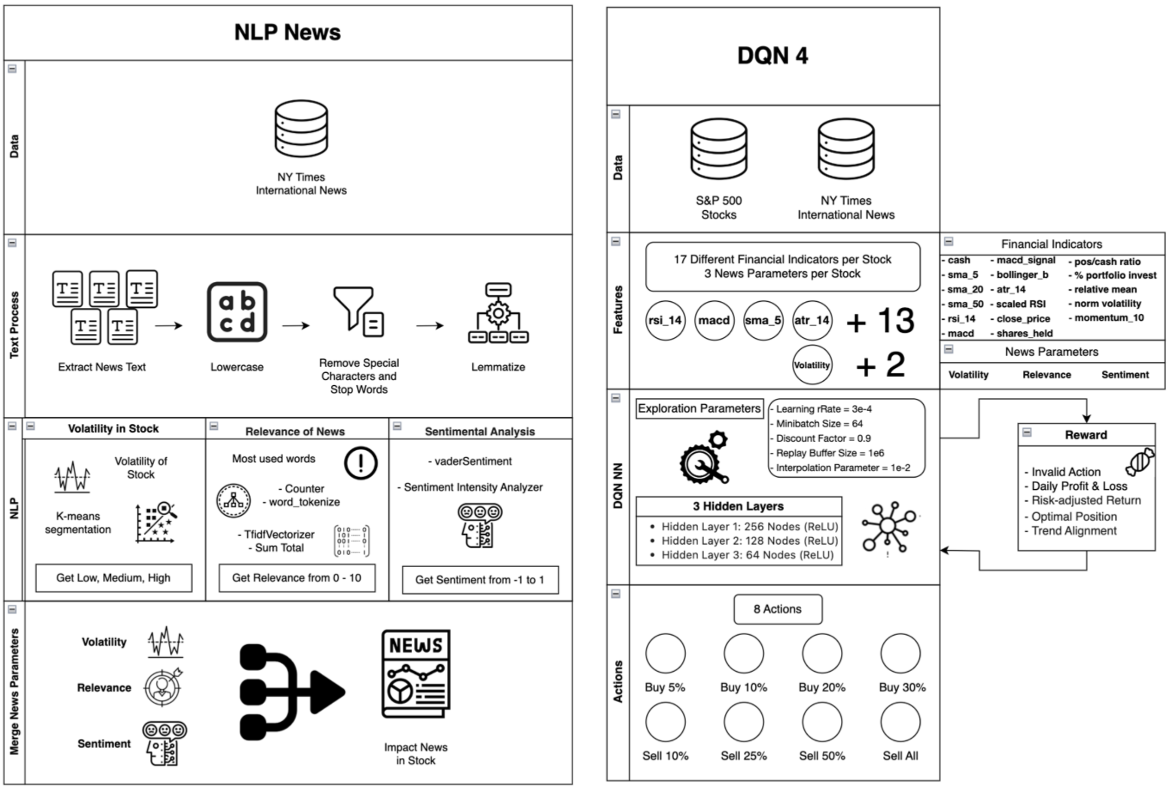

Fig. 3. DQN and NLP News Architecture

DQN and NLP News Model

For this implementation, an NLP process was first carried out to analyze the news and determine how it will impact the stock for trading. In this pipeline, it starts by extracting the text from the news article, and then cleaning it by applying lowercase, removing special characters and stop words, and lemmatizing.

After that, the impact it will have on volatility is calculated. For this, the volatility history of the stock when the news was published (or similar news) is analyzed, and using K-means, a group is assigned: “Low,” “Medium,” or “High.”

To get the relevance of the news to the stock, the process starts by collecting the most directly related news articles from 2000 to 2025. Then, the Python library collections – Counter is used to select the 300 most repeated words in all these articles. After that, TfidfVectorizer is applied to see the frequency and weight of the words in each news item, generating a matrix. Finally, the weights are summed to get a total, and the value is normalized from 0 to 10 to have a defined range.

For the sentiment analysis, the vaderSentiment – SentimentIntensityAnalyzer library was used, applying it to the actual text of the news. This generates a value between -1 to 1, where -1 is a negative sentiment, 0 is neutral, and 1 is positive.

In the implementation of DQN 4, the architecture previously used in DQN 3 (since it had better results) was taken, and these three values were added as parameters in the first input layer to consider the behavior of news in the trading decision.

Implementation

The implementation of the minor contribution was to create a news pipeline that allows reading the news along with a specific stock and can detect the volatility it may cause on it, the relevance of the news topic to the specific stock and the sentiment of the news text, as shown in fig 4. Allowing us to get an idea of how this news is going to impact the stock market

	Date	all_text	Volatility_Category	SPY_Relevance	Sentiment
0	2000-01-03	The Wisdom of Y2K Planning In a welcome anticlimax to two years of dire warnings, the new millennium arrived over the weekend without a computer-driven meltdown of the globe's electronic infrastructure. So far even the countries that did not spend vast sums to prepare for a Y2K-bug disaster have been spared major problems, raising questions about whether all the planning and expense were necessary. The speculation about what might have happened had there been no preparation is intriguing and inevitable. But common sense suggests that inaction was not an option for government and business leaders once the experts determined the existence of a threat to power grids, defense systems, air transport, financial communications and other vital linkages. In a welcome anticlimax to two years of dire warnings, the new millennium arrived over the weekend without a computer-driven meltdown of the globe's electronic infrastructure. So far even the countries that did not spend vast sums to prepare for a Y2K-bug disaster have been spared major problems, raising questions about whether all the planning and expense were necessary. The speculation about what might have happened had there been no preparation is intriguing and inevitable. But common sense suggests that inaction was not an option for government and business leaders once the experts determined the existence of a threat to power grids, defense systems, air transport, financial communications and other vital linkages. Among most computer specialists, the fear of a serious crisis was real. As just about everyone came to learn, their anxiety was that computers would break down because of an inability to recognize the year 2000. A year or so ago, Wired magazine reported that many leading software engineers were themselves planning to spend the New Year's weekend in remote places with stocks of water, cash and food on hand. That fear now seems exaggerated, but it was not entirely fanciful. It reflected the essential fact that even the experts do not fully comprehend the complexity of the systems that now undergird the world's computers. John Koskinen, chairman of the President's Council on Year 2000 Conversion, has warned that it may take weeks or months to be sure that the millennium bug is completely behind us.	Medium	4.04158	-0.99

Fig. 4. News Analysis Pipeline

The implementation of the major contribution was to create a trading simulator where the behavior of the trained neural network can be seen, simulating step by step how it manages the portfolio's money and showing, based on the stock's behavior, whether the portfolio value increased or decreased, as shown in Fig 5. In this simulator, any time period of any stock can be provided, and the neural network starts making financial decisions buying and selling, trying to maximize profit, simulating the real behavior it would have in an actual broker. In this simulator, you can see the initial value of the stock, how its value changed, the portfolio value, and the value it would have if you just invested your money without using the Trading Bot, to can compare results.

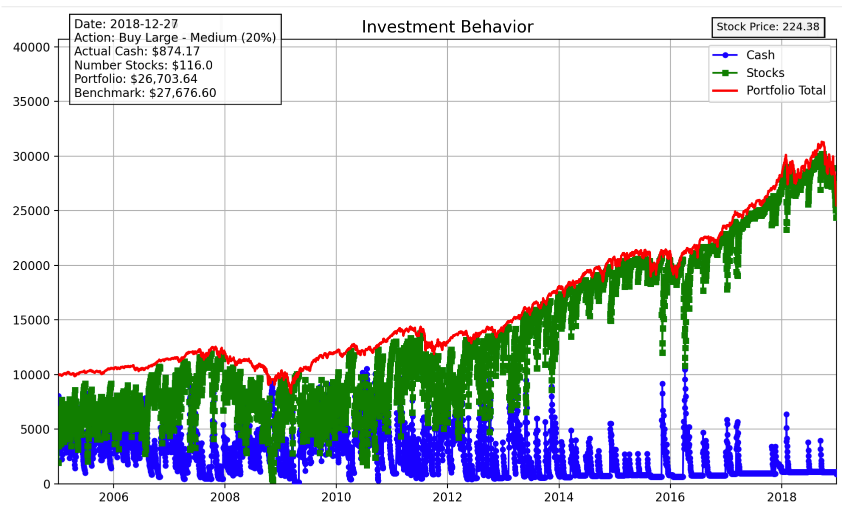

Fig. 5. Simulator

Besides this, the simulator also generates a table with the daily history of the Trading Bot, which can be seen in Fig 6. It shows the time period “Date”, the stock price “Close”, the action taken, the number of shares currently held, the available cash to buy, the money held in stocks, and the total value of the portfolio. With this log, the behavior of the Agent can be analyzed in more detail

Date	Close	v_Close	Prev_Action	Stocks_Held	Actual_Cash	Portfolio_Values	Stocks_Money
2018-12-18	229.378326	229.378326	Buy Large	73.0	304.648308	17049.266136	16744.617828
2018-12-19	225.943192	225.943192	Sell All	0.0	17049.266136	17049.266136	0.000000
2018-12-20	222.265320	222.265320	Buy Large	37.0	8689.368050	16913.184883	8223.816833
2018-12-21	217.711197	217.711197	Buy Large	56.0	4466.326973	16658.153999	12191.827026

Fig. 6. Behavior History

Evaluation and Results

The aim of this experiment is to compare all these different neural network implementations to determine which strategy shows a safer and more stable behavior to increase investments in a financial portfolio.

To do this, we started by comparing the first 3 different DQN architectures (DQN 1, DQN 2, and DQN 3). The S&P 500 stock was selected for these comparison tests, using the years 2000 to 2004 as the training period. In Fig 7, the behaviour during training can be seen, which is when the model explores different possibilities to find relevant patterns for decision-making

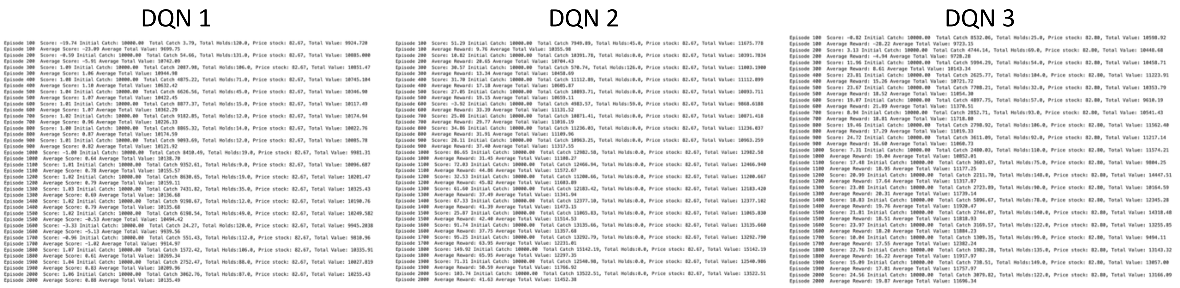

Fig. 7. DQN 1, 2, 3 Training Behavior

During the training period, it can be seen that DQN 1 had problems increasing the portfolio value. Even though it showed different explorations, being limited to buying or selling only one stock did not allow it to increase the investment return by more than 3% in four years, and it obtained an average reward of 1, which is very close to 0 since it kept almost the same value during the whole period.

On the other hand, DQN 2 showed better performance, managing to increase the portfolio value up to 50% in some simulations, giving an average return of 25%, which is significantly higher than DQN 1. However, the reward shown after each iteration was very unstable, showing inconsistent behaviour in each learning implementation, going up to 149 at times and dropping to 25 at others. So, although the performance improved, it did not show safe behaviour when implemented in the simulation.

Lastly, DQN 3 showed more stable and promising results. Although it also managed to increase the portfolio return up to 50%, just like DQN 2, its behaviour was more stable and with more consistent rewards, generating values between 10 and 22. This shows coherence in learning and decision-making, making it the best neural network for trading.

Therefore, after training each of the DQNs, the simulation of their behavior was carried out, and the results obtained are shown in Fig 8.

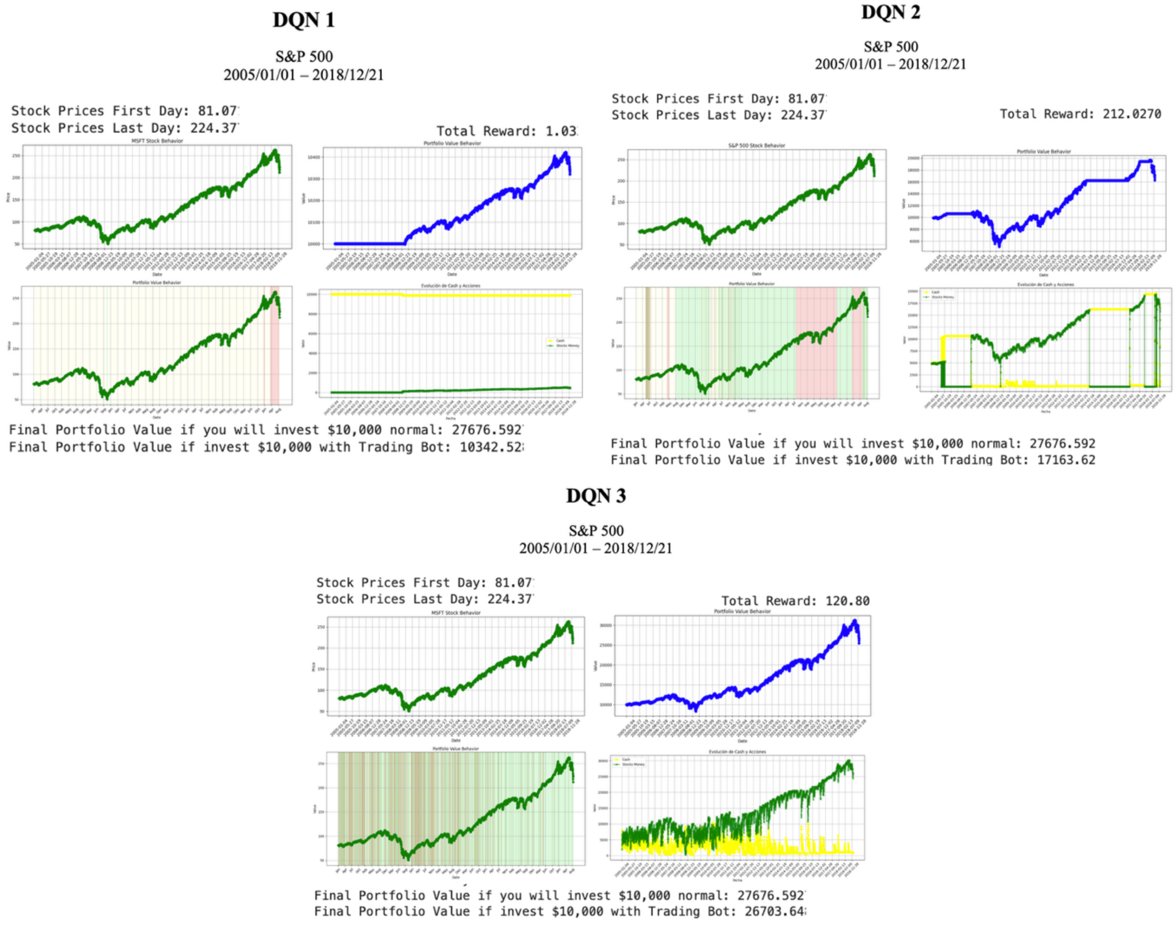

Fig. 8. Simulations of DQNs

To carry out the simulation, the S&P 500 stock was selected, as it was the one used to train the DQNs, covering the period from 2005 to 2018, which follows the training period. All simulations started with an initial investment of $10,000.

As validated during training, DQN 1 achieved a final portfolio value of $10,342 with a reward of 1.03, while DQN 2 reached $17,163 with a reward of 212, and DQN 3 achieved $26,703 with a reward of 120, demonstrating a more consistent and stable performance.

After completing the simulations, DQN 3 was selected as the best-performing model. Therefore, the NLP International News process was integrated into this architecture to create DQN 4, with the goal of evaluating how these models perform under different scenarios.

For this, three distinct one-year periods of the S&P 500 with different market behaviours were selected to observe how both models responded. The chosen periods were 2022–2023 (when stocks declined), 2021–2022 (when stocks rose), and 2015–2016 (when stocks remained stable). The behaviours of these two models can be seen in Fig 9 and 10.

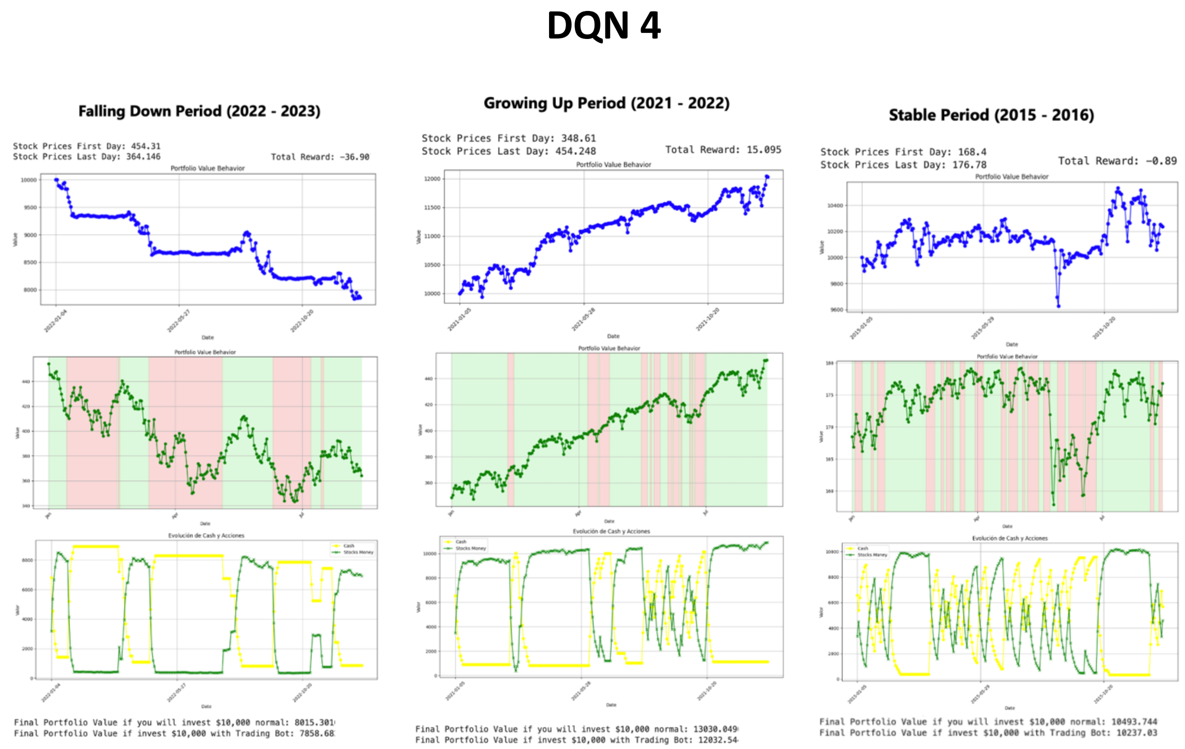

Fig. 9. DQN 4 Performance

The stock price from 2021 to 2022 increased from $348.61 to $454.28, from 2022 to 2023 it dropped from $454.31 to $364.14, and from 2015 to 2016 it remained between $168 and $176. Therefore, if you had invested $10,000 at the beginning of each year, by the end of the 2021–2022 period you would have $13,030.04, in the 2022–2023 period you would have $8,015.30, and in the 2015–2016 period you would have $10,493.74.

The performance of DQN 4 during the 2021–2022 period resulted in a final value of $12,032 with a reward of 15.09. For the 2022–2023 period, it had a final value of $7,858.68 with a reward of -36.90, and during the 2015–2016 period, the final value was $10,237.03 with a reward of -0.89.

DQN 3

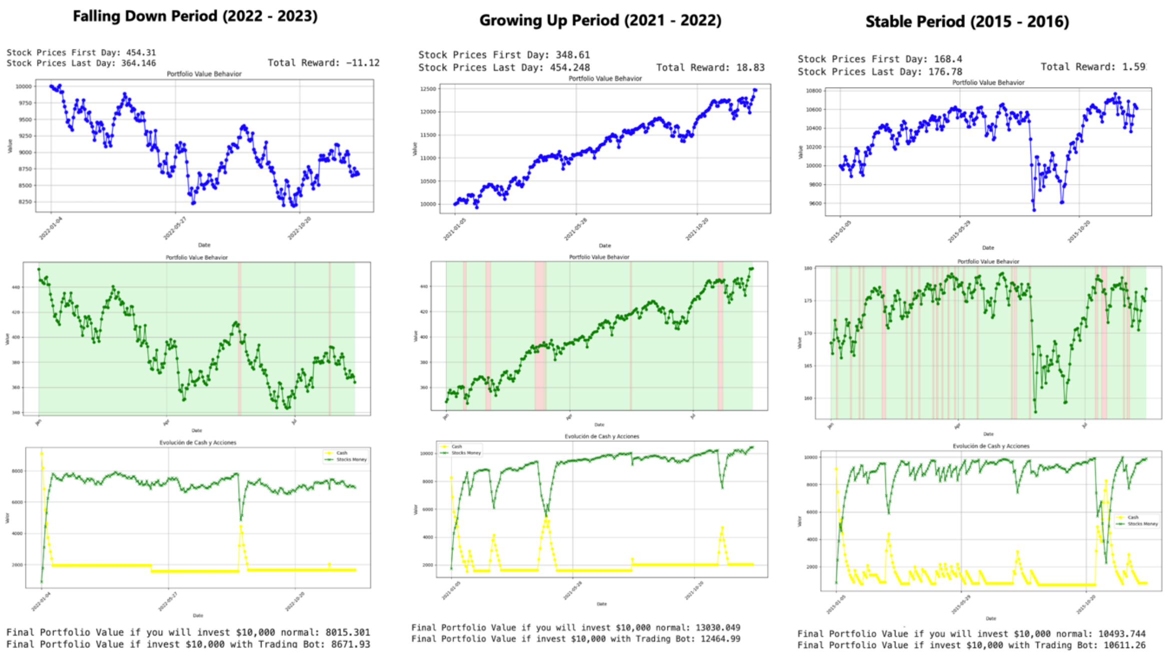

Fig. 10. DQN 3 Performance

And the performance of DQN 3 during the 2021–2022 period resulted in a final value of $12,464.99 with a reward of 18.8. In the 2022–2023 period, it had a final value of $8,671.93 with a reward of -11.12, and in the 2015–2016 period, it achieved a final value of $10,611.26 with a reward of 1.59

Apart from the overall performance obtained from the two models across the three different periods, several evaluation indicators were also generated to assess the behavior of these models. These indicators can be seen in Fig. 11, where, for each model and period, the following metrics are displayed: “Test Period,” “Initial Value,” “Final Value,” “Cumulative Return,” “Maximum Drawdown,” “Sharpe Ratio,” “Profit Factor,” “Win Rate,” and “Final Reward” all presented in a visual format to allow for comparison against the benchmark and determine whether the results were truly positive or negative.

Benchmark

	Test Period	Initial_value	Final_value	Cumulative Return	Maximum Drawdown	Sharpe Ratio
0	Growing Up Period (2021 - 2022)	10000	13030.05	30.30	5.11	2.14
1	Falling Down Period (2022 - 2023)	10000	8015.30	-19.85	24.47	-0.80
2	Stable Period (2015 - 2016)	10000	10493.74	4.94	11.91	0.39

Experiment 3 Results (Only Financial Indicators)

	Test Period	Initial_value	Final_value	Cumulative Return	Maximum Drawdown	Sharpe Ratio	Profit Factor	Win Rate	Final Reward
0	Growing Up Period (2021 - 2022)	10000.0	12464.99	24.65	4.25	2.21	1.43	9.94	18.831776
1	Falling Down Period (2022 - 2023)	10000.0	8671.93	-13.28	18.25	-0.65	0.88	-5.38	-11.129005
2	Stable Period (2015 - 2016)	10000.0	10611.27	6.11	10.59	0.52	1.08	2.46	1.592039

Experiment 5 Results (Adding analysis of international news)

	Test Period	Initial_value	Final_value	Cumulative Return	Maximum Drawdown	Sharpe Ratio	Profit Factor	Win Rate	Final Reward
0	Growing Up Period (2021 - 2022)	10000.0	12032.54	20.33	3.71	1.84	1.38	8.20	15.095241
1	Falling Down Period (2022 - 2023)	10000.0	7858.68	-21.41	21.67	-2.23	0.58	-8.67	-36.907729
2	Stable Period (2015 - 2016)	10000.0	10237.03	2.37	6.50	0.30	1.05	0.96	-0.897639

Fig. 11. Benchmark and DQN 3, DQN 4 Results

In these results, we can see how DQN 3 shows more positive outcomes, as in the 2021–2022 period, when the stock price increased. It was the only period where it performed below the benchmark, with a cumulative return of 24.65 compared to the benchmark's 30.30. However, it still increased the portfolio's value, going from $10,000 at the beginning of the year to $12,464 at the end, which still represents a positive investment return.

On the other hand, in the 2022–2023 and 2015–2016 periods, it outperformed the benchmark by surpassing the cumulative return from -13.28 to -19.85 and 6.11 to 4.94, demonstrating that the trading technique used in this neural network was superior to simple investments in the S&P 500.

Moreover, when comparing DQN 3 with DQN 4, we can see that the results of DQN 3 were superior in every aspect, as the final value was higher in all periods, along with the reward received in each period, demonstrating greater stability and higher performance than DQN 4.

Conclusion and Future Work

The aim of this research was to determine how effective the application of the Deep Q-Network (DQN) algorithm is in neural networks to identify patterns in historical stock market data and to maximize the profits obtained through automated trading strategies.

This research proposes three different neural network architectures implemented with various DQN models, incorporating different financial techniques. The best of these models was then combined with an international news analysis to evaluate the performance of a new DQN model that integrates both financial parameters and news-based insights for trading decision-making, aiming to enhance investment return performance. Results demonstrate that combining international news with financial data within the same neural network is not an effective strategy, as these represent two completely different contextual variables. This can lead the neural network to make biased or confused decisions, ultimately resulting in poorer performance and suboptimal trading actions. On the other hand, the best-performing DQN model showed strong results, outperforming the benchmark in scenarios where the stock market was stable or declining, and only slightly underperforming when the market was rising still yielding positive returns over the initial investment. This highlights it as a solid investment strategy, offering greater security and stability compared to investing solely in the S&P 500.

This research can potentially enhance to all people who don't know how to manage their savings and are afraid of investing, as this would provide reliability and security when investing in specific stocks, encouraging individuals to take better control of their finances and venture into personal investments.

This work can be improved by separating the neural network's intelligence from the NLP analysis of international news, treating them as two separate but complementary processes. First, the news analysis would assess whether it is viable to invest in a particular stock. Once the decision is made regarding whether to invest and how much to invest, the DQN could then be applied exclusively with financial indicators to detect buy/sell patterns and increase investment returns, without being biased by external parameters.

References

[1] Ashraf M, Aslam M, Kakolu S: Artificial Intelligence in Investment Portfolio Optimization: A Comparative Study of Machine Learning Algorithms (2022)

[2] T.V Ambuli, S.Venkatesan, K.Sampath, Kabirdoss Devi and S.Kumaran: AI-Driven Financial Management Optimizing Investment Portfolios through Machine Learning (2024)

[3] Fahima H, Palash S, Musa K, and Hanjala M: Deep Reinforcement Learning for Enhanced Stock Market Prediction with Fine-tuned Technical Indicators (2024)

[4] Mojtaba N, Pooyan N, Hamed J, Shahab S and Amir M: Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis (2020)

[5] Victor C: Application of deep reinforcement learning in stock trading strategies and stock forecasting (2020)

[6] Kumar A and Soam P: A stock market prediction model using Artificial Neural Network (2012)

[7] Chen C, Xuang Y and Caiyi Q: Can natural language processing accurately predict stock market movements based on Reddit World News headlines? (2024)

[8] Kevin T, Jerry N: Natural Language Processing and Multimodal Stock Price Prediction (2023)

[9] Goga S: Challenges of Industry Portfolio Management with Artificial Intelligence (2024)

[10] Jiaozhou Q: Applications and Challenges of AI Technology in Financial Portfolio Management (2024)

[11] Taylan K and Ekrem D: Deep Reinforcement Learning Approach for Trading Automation in the Stock Market (2022)

[12] Junhao Z and Yifei L: Deep Reinforcement Learning for Stock Prediction (2022)

[13] Fahima H, Palash S, Musa K, and Hanjala M: Deep Reinforcement Learning for Enhanced Stock Market Prediction with Fine-tuned Technical Indicators (2024)

[14] Leandro M and Rosangela B: Neural Networks Applied to Stock Market Forecasting: An Empirical Analysis (2010)

[15] Dr. Chandra G, Dr. Menon R, Dr. Sudha M, Yerasuri S, Saha H, Dongol P: The Impact of Artificial Intelligence on Business Strategy and Decision-Making Process (2023)

[16] Eid M, Sagatovna M, Abdurashidov A: The impact of Artificial Intelligence in Decision Making: A comprehensive review (2024)

[17] Lin Chen and Qiang Gao: Application of Deep Reinforcement Learning on Automated Stock Trading (2018)

[18] Oruganti J, Subham N, Varun D and Dammu S: A systematic review of NLP methods for Sentiment classification of Online News Articles (2023)

[19] Caio M, Gullal S and Gaurish T: Combining sentiment analysis classifiers to explore multilingual news articles covering London 2012 and Rio 2016 Olympics (2022)

[20] Fred M, Liang W, Uraz Y and Sthepen R: Identifying Framing Bias in Online News (2018)

[21] New York Time News Dataset https://www.kaggle.com/datasets/aryansingh0909/nyt-articles-21m-2000-present

[22] Yahoo Finance API https://python-yahoofinance.readthedocs.io/en/latest/api.html

[23] Presentation Video Presentation_Video_Rodrigo_Esparza.mp4

[24] Code Video Code_Video_Rodrigo_Esparza.mp4