Knowledge-Based Systems 284 (2024) 111290

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Deep LSTM and LSTM-Attention Q-learning based reinforcement learning in oil and gas sector prediction

David Opeoluwa Oyewola [a,*], Sulaiman Awwal Akinwunmi [a], Temidayo Oluwatosin Omotehinwa [b]

[a] Department of Mathematics and Statistics, Federal University Kashere, Gombe State, Nigeria
[b] Department of Mathematics and Computter Science, Federal University of Health Science, Otukpo, Nigeria

ARTICLE INFO

Keywords:
Stock market
Prediction
Markov decision process
Deep learning
Long short term memory
Attention mechanism

ABSTRACT

Accurate prediction of stock market trends and movements holds great significance in the financial industry as it enables investors, traders, and decision-makers to make informed choices and optimize their investment strategies. In the context of the oil and gas sector, where stock prices are influenced by complex market dynamics and various external factors, reliable predictions are essential for effective decision-making and risk management. This study proposes Deep Long Short-Term Memory Q-Learning (DLQL) and Deep Long Short-Term Memory Attention Q-Learning (DLAQL) models and state-of-the-art Long Short-Term Memory (LSTM) for predicting stock prices in the oil and gas sector. The study utilizes historical stock price data of Cenovus Energy Inc. (CVE), MPLX LP (MPLX), Cheniere Energy Inc. (LNG), and Suncor Energy Inc. (SU) to create and validate these models. The research employs the Markov Decision Process (MDP) framework, a widely-used reinforcement learning technique, to train the deep LSTM Q-Learning and deep LSTM Attention Q-Learning models. This framework allows the models to learn optimal policies based on historical data, enabling them to make accurate predictions and adapt to changing market conditions. The findings of this study reveal that the proposed DLQL and DLAQL perform excellently well in terms of prediction accuracy in the oil and gas sector. The inclusion of attention mechanisms in the DLAQL model further enhances its performance by allowing it to focus on important features and capture relevant information. The results of this research underscore the potential of deep LSTM Q-Learning and deep LSTM Attention Q-Learning models in stock market prediction within the oil and gas sector. The application of these models can lead to improved decision-making, enhanced risk management, and increased profitability for market participants. Further exploration and refinement of these models, along with the incorporation of additional variables and market indicators, can contribute to the development of more sophisticated prediction models in the future. Overall, this study contributes to the advancement of stock market prediction techniques, specifically in the oil and gas sector, by introducing and evaluating the efficacy of deep LSTM Q-Learning and deep LSTM Attention Q-Learning models. The findings highlight the importance of accurate stock market predictions and demonstrate the potential benefits of leveraging these models within the MDP framework to support decision-making and risk management in the dynamic and competitive oil and gas industry.

1. Introduction

Stock market prediction [1–5] is a topic of immense interest and significance in the financial industry. Accurately forecasting stock prices [6–8] and market trends is crucial for investors, traders, and decision-makers as it provides valuable insights to make informed choices, optimize investment strategies [9–12], and mitigate risks. In particular, the oil and gas sector [13], known for its complex market dynamics and susceptibility to various external factors, heavily relies on accurate predictions to drive effective decision-making and risk management strategies. The emergence of advanced technologies and machine learning techniques has revolutionized stock market prediction [14–20], enabling more sophisticated and accurate forecasting models. Deep learning models, such as Recurrent Neural Networks [21],

[*] Corresponding author.
E-mail address: davidakaprof01@yahoo.com (D.O. Oyewola).

https://doi.org/10.1016/j.knosys.2023.111290
Received 27 September 2023; Received in revised form 1 December 2023; Accepted 7 December 2023
Available online 8 December 2023

Convolutional Neural Networks [22,23] and Long Short-Term Memory (LSTM) [24–30] networks, have shown great promise in capturing complex patterns and dependencies in time series data, making them well-suited for predicting stock prices. Deep learning has gained significant attention due to its ability to handle sequential data and capture long-term dependencies, which are essential in financial time series analysis. These models have demonstrated impressive performance in various domains, including natural language processing [31], speech recognition [32], and, notably, stock market prediction.

Reinforcement learning techniques [33–40], on the other hand, have been widely utilized to optimize decision-making processes in dynamic environments. One popular reinforcement learning algorithm is Q-Learning, which employs trial-and-error exploration to learn the optimal actions that maximize rewards. By combining deep learning models, such as LSTM networks, with reinforcement learning techniques like Q-Learning, researchers have aimed to develop more accurate and effective approaches to stock market prediction. The oil and gas sector presents unique challenges in terms of stock market prediction. The industry is highly influenced by geopolitical events, supply-demand dynamics, regulatory changes, and environmental factors, which necessitate sophisticated prediction models that can account for the sector's inherent complexities and uncertainties. Accurate stock market predictions in the oil and gas sector are crucial for stakeholders to navigate the market landscape successfully and make well-informed investment decisions. Deep learning models, such as Long Short-Term Memory (LSTM) [41–45] networks, have shown promise in capturing intricate patterns and dependencies in time series data. LSTM networks are well-suited for stock market prediction as they excel at modeling long-term dependencies and handling sequential data effectively.

LSTM can suffer from vanishing and exploding gradient problems, which affect their ability to capture long-range dependencies in data. Training deep LSTM with a large number of time steps can lead to gradients becoming too small (vanishing) or too large (exploding), making it difficult to learn from distant past information. Attention mechanisms, when integrated with LSTM, can focus on specific parts of the input sequence that are most relevant at each time step. This attention mechanism allows the model to assign different weights to different elements of the input, effectively capturing complex feature interactions. It helps in giving more importance to critical information within the sequence, which is particularly valuable in the oil and gas sector for identifying factors with significant impact.

Reinforcement learning techniques, particularly Q-Learning, have been extensively used to optimize decision-making in dynamic environments. However, to the best of our knowledge, no study in the literature has explored the integration of LSTM, LSTM attention mechanism, and Q-Learning for stock market prediction. In this paper, we present an innovative approach that combines the power of LSTM, LSTM attention mechanism, and Q-Learning, referred to as Deep LSTM Q-Learning (DLQL) and Deep LSTM Attention Q-Learning (DLAQL). This integration aims to enhance prediction accuracy and optimize decision-making processes in the oil and gas sector. By leveraging deep learning's ability to capture complex patterns and dependencies, attention mechanisms' focus on relevant information, and reinforcement learning's adaptability to changing market conditions, we propose novel models tailored specifically for stock market prediction in the oil and gas sector.

The major contributions of this work are as follows:

1. We propose Deep LSTM Q-Learning (DLQL) and Deep Learning LSTM Attention Q-Learning (DLAQL) models that combine the strengths of LSTM networks and attention mechanisms. These models effectively capture long-term dependencies and focus on relevant information, improving prediction accuracy and interpretability in the oil and gas sector.
2. We employ deep Q-Learning, a reinforcement learning technique, to optimize decision-making processes in the dynamic stock market environment. By learning from trial and error and optimizing policies based on rewards and penalties, our models adapt to changing market conditions and make informed trading decisions.
3. We evaluate the proposed DLQL and DLAQL models using real-world data from prominent oil and gas companies, including Cenovus Energy Inc. (CVE), MPLX LP (MPLX), Cheniere Energy Inc. (LNG), and Suncor Energy Inc. (SU). The evaluation provides insights into the effectiveness of our models and their potential applications in the oil and gas sector.
4. This study contributes to the existing body of research on stock market prediction by introducing innovative models specifically designed for the oil and gas sector. By addressing the unique market dynamics and external factors of this sector, our work offers valuable insights and advances the understanding of stock market prediction in the oil and gas industry.

In the following sections, we review the related works, whereas in the methodology section, we present a detailed description of the proposed DLQL and DLAQL models, including their architecture, training process, and decision-making mechanisms. We then present the experimental setup, including the dataset, evaluation metrics, and performance comparison. The results and analysis section provides an in-depth evaluation of our models' performance, highlighting their accuracy and effectiveness in predicting stock prices in the oil and gas sector. Finally, we conclude the paper by summarizing the contributions, discussing the implications of our findings, and suggesting avenues for future research in stock market prediction.

2. Related works

In recent years, there has been a growing interest in the application of deep learning and reinforcement learning techniques to stock market prediction. These techniques have shown promise in capturing the complex temporal dependencies in stock price data, which can be difficult to model using traditional statistical methods. One of the most popular deep learning techniques for stock market prediction is the long short-term memory (LSTM) network. LSTM networks are capable of learning long-range dependencies, which is essential for accurately forecasting stock prices. In a recent study, Lakshminarayanan et al. [41] compared the performance of LSTM networks with support vector machine (SVM) regression models for stock market prediction. They found that LSTM networks outperformed SVM models in all scenarios, with a significant reduction in root mean squared error (RMSE). Another promising approach to stock market prediction is reinforcement learning. Reinforcement learning algorithms learn to make decisions by trial and error, in an environment where the consequences of their actions are rewarded or penalized. This approach is effective for trading in financial markets, as it allows the algorithm to learn from its own mistakes and improve its performance over time. In a recent study, Baek et al. [46] proposed a novel reinforcement learning algorithm for stock market prediction. Their algorithm, called ModAugNet, uses a combination of LSTM networks and reinforcement learning to learn to trade stocks profitably. They evaluated the performance of ModAugNet on two different stock market datasets and found that it outperformed several baseline algorithms. These studies suggest that deep learning and reinforcement learning techniques have the potential to improve the accuracy of stock market prediction. However, more research is needed to develop robust and scalable models that can be used in practice.

Reinforcement learning (RL) has been employed in stock market prediction in recent years. RL algorithms learn to make decisions by trial and error, in an environment where the consequences of their actions are rewarded or penalized. This approach is effective for trading in financial markets, as it allows the algorithm to learn from its own mistakes and improve its performance over time. One popular RL algorithm for stock market prediction is deep Q-learning (DQN). DQN is a type of RL algorithm that uses a neural network to approximate the value function, which is a function that maps from states to expected rewards.

DQN is effective for stock market prediction and has outperformed traditional statistical methods in some studies. In [47], the authors proposed adaptive stock trading strategies with deep reinforcement learning named GDQN (Gated Deep Q-learning trading strategy) and GDPG (Gated Deterministic Policy Gradient trading strategy). They tested GDQN and GDPG in both trending and volatile stock markets from different countries to verify their robustness and effectiveness. The experimental results showed that the proposed GDQN and GDPG not only outperformed the Turtle trading strategy but also achieved more stable returns than a state-of-the-art direct reinforcement learning method, DRL trading strategy, in the volatile stock market. In another study, Théate et al. [48] proposed a novel deep reinforcement learning (DRL) named Trading Deep Q-Network (TDQN) trading policy to maximize the resulting Sharpe ratio performance indicator on a broad range of stock markets. The TDQN trading policy was evaluated on a variety of stock markets and was found to outperform several baseline algorithms. Similarly, Li et al. [49] proposed a new deep reinforcement learning model to implement stock trading. This model analyzes the stock market through stock data, technical indicators, and candlestick charts, and learns dynamic trading strategies. The agent in reinforcement learning makes trading decisions based on the state of the stock market, which is fused with the features of different data sources extracted by the deep neural network. Experiments on the Chinese stock market dataset and the S&P 500 stock market dataset showed that the proposed trading strategy can obtain higher profits compared with other trading strategies. These studies suggest that RL approaches have the potential to improve the accuracy of stock market prediction. However, more research is needed to develop robust and scalable models that can be used in practice.

In a recent study, Kabbani et al. [50] developed a deep reinforcement learning (DRL) model to generate profitable trades in the stock market. They formulated the trading problem as a Partially Observed Markov Decision Process (POMDP) model, considering the constraints imposed by the stock market, such as liquidity and transaction costs. They then solved the formulated POMDP problem using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, reporting a 2.68 Sharpe Ratio on test data. The results of this study show the superiority of DRL in financial markets over other types of machine learning. This is because DRL can learn from experience and adapt to changing market conditions. Additionally, DRL can take into account the constraints of the stock market, such as liquidity and transaction costs. Several studies have explored the application of deep reinforcement learning (DRL) in stock trading strategy optimization and have yielded promising results. In one study [51], a Convolutional Neural Network (CNN) was utilized within a Double Deep Q-Network (DDQN) to analyze the S&P 500 Index returns. The DDQN, trained and tested on the 30 largest stocks in the index, outperformed the index returns and provided insights into the neural network's attention on candlestick images using feature map visualizations. In another investigation [52], deep reinforcement learning was employed for stock trading strategy and investment decisions in the Indian market. Three classical DRL models, including Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Dueling Double Deep Q-Network (D3QN), were systematically evaluated on ten Indian stock datasets. The performance of these models was compared, demonstrating the potential of DRL in stock trading strategy optimization. Moreover, a novel trading agent based on deep reinforcement learning was proposed [53], aiming to autonomously make trading decisions and generate profits in dynamic financial markets. The agent extended the value-based deep Q-network (DQN) and asynchronous advantage actor-critic (A3C) models to better adapt to the trading market. Various mechanisms, such as position-controlled action and n-step reward, were incorporated to enhance the agent's practicality in real trading environments. Experimental results showed that the proposed trading agent outperformed baseline methods and achieved stable risk-adjusted returns in both stock and futures markets. In the context of pairs trading, a hybrid deep reinforcement learning method called HDRL-Trader was introduced [54]. This approach employed two independent reinforcement learning networks to determine trading actions and stop-loss boundaries. The HDRL-Trader model incorporated novel techniques, including dimensionality reduction, clustering, regression, behavior cloning, prioritized experience replay, and dynamic delay. Experimental results on twenty stock pairs in the Standard & Poor's 500 index demonstrated that HDRL-Trader achieved significant positive return rates, outperforming other state-of-the-art reinforcement learning methods for pairs trading.

Furthermore, deep reinforcement learning was explored for optimizing stock trading strategy and maximizing investment return [55]. The study selected 30 stocks as trading assets and used their daily prices as training and trading market environments. A deep reinforcement learning agent was trained to develop an adaptive trading strategy. The agent's performance was evaluated and compared with the Dow Jones Industrial Average and a traditional min-variance portfolio allocation strategy. The results indicated that the proposed deep reinforcement learning approach outperformed the baselines in terms of both the Sharpe ratio and cumulative returns. Similarly, a multi-agent deep reinforcement learning framework was proposed [56], which leveraged the collective intelligence of multiple agents, each specialized in trading on a specific timeframe. The framework operated in a hierarchical structure, with knowledge flowing from higher timeframe agents to lower timeframe agents, ensuring robustness to noise in financial time series. The Deep Q-learning algorithm was employed to train the agents, and extensive numerical experiments were conducted on a historical dataset of the EUR/USD currency pair. The proposed framework outperformed single independent agents and benchmark trading strategies across all investigated trading timeframes, as evaluated by several return-based and risk-based performance measures. Table 1 is the list of related works.

3. Methodology

The research proposes the use of Deep LSTM Attention Q-Learning and LSTM methods in the context of stock market trading. These methods leverage deep neural networks, LSTM layers, attention mechanisms, and Q-Learning algorithms to enhance the agent's decision-making process. The stock market environment is modeled as a Markov Decision Process (MDP), where the agent aims to maximize cumulative rewards by buying or selling stocks. The agent's actions are guided by the Q-values predicted by the DQN, and rewards are computed based on the differences between buying and selling prices. The proposed methods aim to capture temporal dependencies, identify important features, and improve the agent's performance in stock market trading.

3.1. Stock market environment

The stock market environment considered in this study comprises the stock prices of Cenovus Energy Inc. (CVE), MPLX LP (MPLX), Cheniere Energy Inc. (LNG), and Suncor Energy Inc. (SU). These companies represent a diverse range of industries and provide valuable insights into the dynamics of the stock market. By analyzing the stock prices of these companies, the agent gains access to a comprehensive information source, enabling the identification of market trends, patterns, and potential opportunities.

3.2. Agent

The role of the agent is pivotal in the reinforcement learning framework as it serves as the primary entity that interacts with the stock market environment. This research paper explores the various aspects of the agent's functioning, including initialization, Q-network model, epsilon-greedy strategy, and experience replay. During the initialization phase, the agent configures important parameters such as the size of the state and action spaces, memory buffer capacity, and inventory

Table 1
List of related works.

References	Proposed techniques	Contributions
[41]	Proposed LSTM networks with support vector machine (SVM) regression models for stock market prediction.	They found that LSTM networks outperformed SVM models in all scenarios, with a significant reduction in root mean squared error (RMSE).
[46]	Proposed a novel reinforcement learning algorithm for stock market prediction. Their algorithm, called ModAugNet, uses a combination of LSTM networks and reinforcement learning to learn to trade stocks profitably.	They evaluated the performance of ModAugNet on two different stock market datasets, and found that it outperformed a number of baseline algorithms.
[47]	Proposed adaptive stock trading strategies with deep reinforcement learning named GDQN (Gated Deep Q-learning trading strategy) and GDPG (Gated Deterministic Policy Gradient trading strategy).	They tested GDQN and GDPG in both trending and volatile stock markets from different countries to verify their robustness and effectiveness. The experimental results showed that the proposed GDQN and GDPG not only outperformed the Turtle trading strategy, but also achieved more stable returns than a state-of-the-art direct reinforcement learning method, DRL trading strategy, in the volatile stock market.
[48]	Proposed a novel deep reinforcement learning (DRL) named Trading Deep Q-Network (TDQN) trading policy to maximize the resulting Sharpe ratio performance indicator on a broad range of stock markets.	The TDQN trading policy was evaluated on a variety of stock markets, and was found to outperform a number of baseline algorithms.
[49]	Proposed a new deep reinforcement learning model to implement stock trading. This model analyzes the stock market through stock data, technical indicators, and candlestick charts, and learns dynamic trading strategies. The agent in reinforcement learning makes trading decisions based on the state of the stock market, which is fused with the features of different data sources extracted by the deep neural network.	Experiments on the Chinese stock market dataset and the S&P 500 stock market dataset showed that the proposed trading strategy can obtain higher profits compared with other trading strategies. These studies suggest that RL approaches have the potential to improve the accuracy of stock market prediction. However, more research is needed to develop robust and scalable models that can be used in practice
[50]	Developed a deep reinforcement learning (DRL) model to generate profitable trades in the stock market. They formulated the trading problem as a Partially Observed Markov Decision Process (POMDP) model, considering the constraints imposed by the stock market, such as liquidity and transaction costs.	They solved the formulated POMDP problem using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, reporting a 2.68 Sharpe Ratio on test data. The results of this study show the superiority of DRL in financial markets over other types of machine learning.
[51]	Proposed a Convolutional Neural Network (CNN) was with a Double Deep Q-Network (DDQN) to analyze the S&P 500 Index returns.	The DDQN, trained and tested on the 30 largest stocks in the index, outperformed the index returns and provided insights into the neural network's attention on candlestick images using feature map visualizations.
[52]	Deep reinforcement learning was employed for stock trading strategy and investment decisions in the Indian market. Three classical DRL models, including Deep Q-Network (DQN), Double Deep Q-Network (DDQN), and Dueling Double Deep Q-Network (D3QN), were systematically evaluated on ten Indian stock datasets.	The performance of these models was compared, demonstrating the potential of DRL in stock trading strategy optimization.
[53]	Proposed a novel trading agent based on deep reinforcement learning, aiming to autonomously make trading decisions and generate profits in dynamic financial markets. The agent extended the value-based deep Q-network (DQN) and asynchronous advantage actor-critic (A3C) models to better adapt to the trading market. Various mechanisms, such as position-controlled action and n-step reward, were incorporated to enhance the agent's practicality in real trading environments.	Experimental results showed that the proposed trading agent outperformed baseline methods and achieved stable risk-adjusted returns in both stock and futures markets.
[54]	Proposed a hybrid deep reinforcement learning method called HDRL-Trader. This approach employed two independent reinforcement learning networks to determine trading actions and stop-loss boundaries. The HDRL-Trader model incorporated novel techniques, including dimensionality reduction, clustering, regression, behavior cloning, prioritized experience replay, and dynamic delay.	Experimental results on twenty stock pairs in the Standard & Poor's 500 index demonstrated that HDRL-Trader achieved significant positive return rates, outperforming other state-of-the-art reinforcement learning methods for pairs trading.
[55]	Deep reinforcement learning was explored for optimizing stock trading strategy and maximizing investment return. The study selected 30 stocks as trading assets and used their daily prices as training and trading market environments. A deep reinforcement learning agent was trained to develop an adaptive trading strategy. The agent's performance was evaluated and compared with the Dow Jones Industrial Average and a traditional min-variance portfolio allocation strategy.	The results indicated that the proposed deep reinforcement learning approach outperformed the baselines in terms of both the Sharpe ratio and cumulative returns.
[56]	Proposed a multi-agent deep reinforcement learning framework, which leveraged the collective intelligence of multiple agents, each specialized in trading on a specific timeframe. The framework operated in a hierarchical structure, with knowledge flowing from higher timeframe agents to lower timeframe agents, ensuring robustness to noise in financial time series. The Deep Q-learning algorithm was employed to train the agents, and extensive numerical experiments were conducted on a historical dataset of the EUR/USD currency pair.	The proposed framework outperformed single independent agents and benchmark trading strategies across all investigated trading timeframes, as evaluated by several return-based and risk-based performance measures

management. These parameters provide the agent with the necessary resources and settings to operate efficiently within the stock market environment. The agent employs a deep neural network model, constructed using the Keras library, to approximate the Q-values of state-action pairs. This model architecture incorporates LSTM layers, enabling the agent to capture and leverage temporal dependencies present in the stock price data. Additionally, the attention mechanism further enhances the agent's ability to focus on relevant information within the data. To balance exploration and exploitation during the learning process, the agent adopts the epsilon-greedy strategy for action selection. Based on the current state and the predicted Q-values from the model, the agent either explores the environment by selecting a random action with a certain probability (epsilon) or exploits the learned knowledge by selecting the action with the highest Q-value. Experience replay is employed to enhance training stability and sample efficiency. Past experiences, comprising tuples of state, action, reward, and done, are stored in a memory buffer. During training, a random batch of experiences is sampled from the buffer, breaking the temporal correlation between consecutive samples. This technique improves the agent's learning efficiency by utilizing a diverse range of experiences. By utilizing LSTM layers, attention mechanisms, the epsilon-greedy strategy, and experience replay, the agent effectively learns and adapts within the stock market environment. The LSTM layers capture long-term dependencies and patterns, while attention mechanisms enhance information processing. The epsilon-greedy strategy balances exploration and exploitation, allowing the agent to explore new actions while exploiting learned knowledge. Experience replay facilitates efficient learning by leveraging past experiences. Throughout the training process, the agent iteratively interacts with the stock market environment, updates the Q-network's weights using Q-learning, and refines its decision-making strategies. By leveraging past experiences and incorporating feedback from the environment, the agent gradually improves its ability to make informed decisions and maximize profit.

3.3. States

The state plays a vital role in the reinforcement learning framework, serving as a key component that provides the agent with essential information about the current state of the environment. This research paper delves into the significance of the state and its construction methodology within the context of stock market analysis. The state vector is carefully designed to capture relevant features and context necessary for the agent's decision-making process. It is constructed as a sequence of sigmoid-transformed differences between consecutive stock prices. These differences represent the changes in stock prices over time, allowing the agent to perceive the direction and magnitude of price movements. The application of the sigmoid transformation normalizes and compresses the differences within a suitable range, ensuring that the state values are scaled appropriately for the neural network to process effectively. In addition to the sigmoid-transformed differences, the state representation incorporates historical price changes. By including the past behavior of stock prices, the state enables the agent to gain insights into the temporal dynamics of the market. This historical perspective facilitates the identification of patterns, trends, and potential dependencies that may influence future stock prices. By considering historical price changes, the state representation provides a contextual view of how stock prices have evolved over a specific period, enhancing the agent's understanding of the market dynamics. The combination of sigmoid-transformed differences and historical price changes in the state representation empowers the agent with valuable insights into the stock market environment. It equips the agent with information about the direction and magnitude of price changes, enabling it to identify trends, patterns, and potential opportunities. By considering the temporal context of stock price movements, the state representation enhances the agent's decision-making process, allowing it to make informed choices that maximize its performance within the dynamic stock market landscape. The mathematical equation of states is given as:

$$S_t = \left[c_t^i,\ i\ \in \aleph\right] \tag{1}$$

Where S_t is the state vector at each time step t and c_t^i is the closing price of assets i in $\aleph$ at time step t.

3.4. Action space

Action space plays a crucial role in the context of reinforcement learning, enabling the agent to interact with the environment and shape its future outcomes. This research paper highlights the significance of actions within the reinforcement learning process and their application in the specific domain of stock market trading. The agent operates within a discrete action space, which consists of three distinct actions: "hold," "buy," and "sell." Each action represents a decision that the agent can make at a particular state within the environment. The selection of these specific actions is tailored to the problem domain and aligns with the desired behavior of the agent. In the case of stock market trading, these actions correspond to decisions related to holding onto stocks, buying new stocks, or selling existing stocks. To guide the decision-making process, the agent relies on Q-values predicted by the Deep Q-Network (DQN) model. The Q-values serve as estimations of the future rewards associated with each action in a given state. By evaluating the Q-values, the agent can assess the potential outcomes of different actions and determine the action that maximizes the expected future rewards. The action selection process is typically guided by an exploration-exploitation strategy, such as epsilon-greedy, which strikes a balance between exploring new actions and exploiting the learned knowledge. Using the Q-values as a reference, the agent makes decisions on whether to hold, buy, or sell stocks. The decision-making process involves selecting the action with the highest Q-value in the current state. In cases where multiple actions have identical Q-values, tie-breaking strategies can be employed to resolve the ambiguity. Through this action selection mechanism, the agent adapts its behavior and learns an optimal policy that maximizes cumulative rewards over time. By continuously interacting with the environment and updating the Q-values based on feedback, the agent refines its decision-making process and improves its performance. By defining a specific set of actions and leveraging the Q-values predicted by the DQN model, the agent gains the ability to make decisions that directly influence its interaction with the environment. The action selection process empowers the agent to navigate the decision space, explore different strategies, and ultimately learn to choose actions that lead to favorable outcomes in the context of stock market trading. The mathematical equation of action space is given as:

$$A_t = \left\{a_t^i \middle|\ i\ \in\ \aleph\right\} = \left\{a_t^o,\ a_t^1,\ \ldots,\ a_t^{\aleph}\right\} \tag{2}$$

subject to

$$a_t^i\ \in\ \mathbb{Z}^{\aleph} \tag{3}$$

$$-k_{max} \leq a_t^i\ \leq k_{max},\ \forall\ i\ \in\ \aleph \tag{4}$$

$$a_t^i = h_t^i \text{ if } \left|a_t^i\right| > h_t^i,\ \forall\ a_t\ \in \mathbb{Z} \tag{5}$$

Where $\aleph$ is the four assets considered in this analysis, A_t is the action vector sent by the agent to the environment, a_t^i is the action to buy/sell/hold for asset i at time step t, k_{max} is the maximum number of shares the agent can reallocate of an individual asset at each time step t, h_t^i is the number of shares of assets i at time step t.

3.5. Rewards function

Rewards function plays a fundamental role within the reinforcement learning framework, serving as a crucial component that provides feedback to the agent and guides its learning process. This research paper delves into the significance of rewards in the context of stock market trading, highlighting their computation, the calculation of profit, and their role in the agent's learning process. The computation of rewards occurs when the agent chooses the "buy" action and purchases a stock at the current price. The reward associated with this action is determined based on the subsequent selling price and the initial buying price. It quantifies the profit or loss generated from the trade.

Specifically, the reward is computed by subtracting the buying price from the current stock price. If the result is positive, it signifies a profit. Conversely, if the result is zero or negative, it indicates no profit or loss. This calculation ensures that rewards accurately reflect the outcomes of the agent's trading decisions. To track and update the agent's performance, the total profit obtained throughout the trading process is continuously monitored. The cumulative profit is computed as the sum of individual trade profits. As the agent sells stocks at higher prices, the profit increases accordingly. This mechanism enables the agent to keep a running record of the accumulated profit, providing a measure of its success in generating positive returns from its trades. Rewards serve as a vital feedback mechanism for the agent, offering insights into the quality of its actions. Positive rewards indicate successful and profitable trades, reinforcing the agent's decision-making process. On the other hand, negative rewards indicate losses or unfavorable outcomes, guiding the agent to refine its strategies. The objective of the agent is to maximize its cumulative rewards across multiple episodes, aiming to achieve higher profitability. By receiving rewards based on profitable trades, the agent learns to recognize patterns, trends, and indicators that lead to favorable outcomes. Using the rewards and the Q-learning algorithm, the agent updates its Q-network, establishing associations between actions and the expected future rewards. This allows the agent to make more informed decisions in similar situations, leveraging its learned knowledge to maximize rewards. Through its interaction with the environment and the feedback from rewards, the agent progressively refines its trading strategies, identifying profitable opportunities, and adapting its actions accordingly. The mathematical equations are given as:

$$R(S_t, a_t^i) = p_t - p_{t-1} \tag{6}$$

$$p_t = h_t c_t - h_t(c_{t-1} o d_t) \tag{7}$$

$$d_t = \begin{cases} d_{buy}, \, if \; a_t^i > 0 \\ 0, \; if \; a_t^i = 0 \\ d_{sell}, \; if \; a_t^i < 0 \end{cases} \tag{8}$$

Where h_t^i is the number of shares of assets i at time step t, c_t^i is the closing price of assets i at time step t, c_{t-1} is the closing price of the previous period.

3.6. *Proposed deep Q-Learning*

In this study, we propose two novel reinforcement learning techniques for stock market trading: Deep LSTM Attention Q-Learning and Deep LSTM Q-Learning. These techniques combine deep learning models, specifically LSTM (Long Short-Term Memory) networks, with Q-Learning, a popular reinforcement learning algorithm. Fig. 1 displays the Schematic diagram of deep Long Short-Term Memory (LSTM) and Long Short Term Memory (LSTM)-Attention Q-Learning Based Reinforcement Learning

3.6.1. *Deep LSTM Attention Q-Learning (DLAQL)*

Deep LSTM Attention Q-Learning incorporates an attention mechanism into the traditional Q-Learning framework. The attention mechanism enables the agent to focus on relevant information within the state

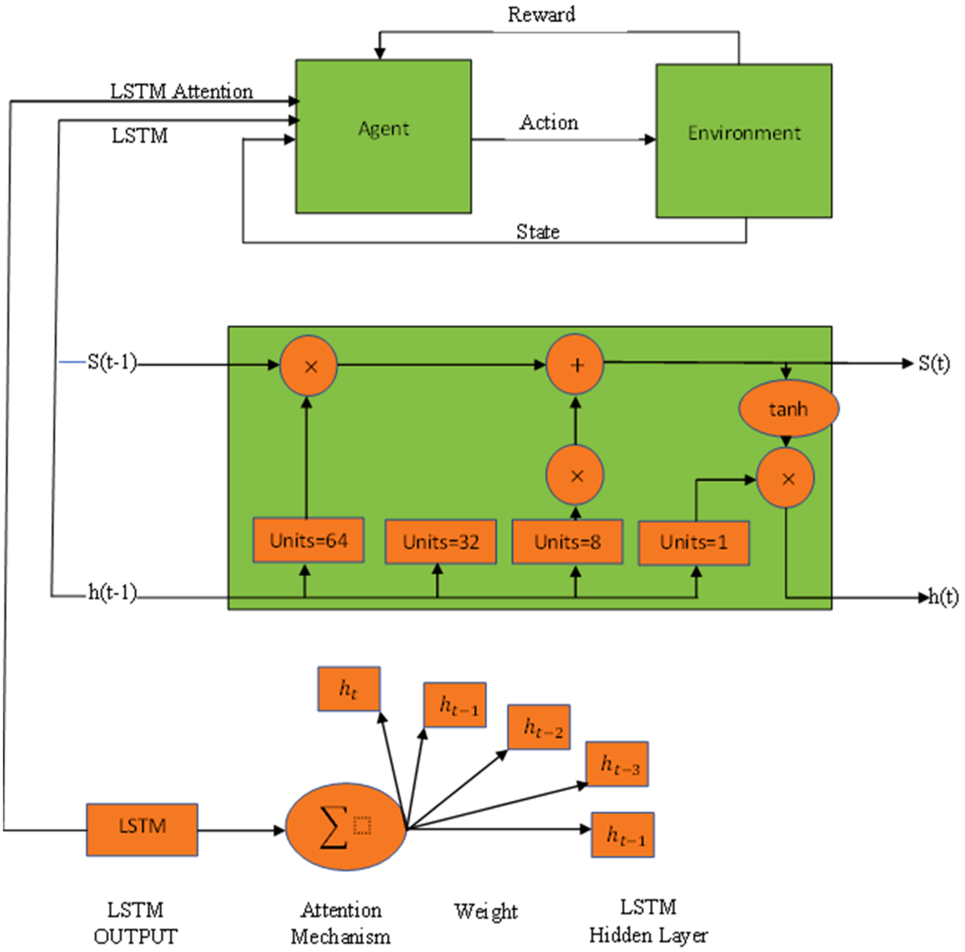

Fig. 1. The Schematic diagram of deep Long Short-Term Memory (LSTM) and Long Short Term Memory (LSTM)-Attention Q-Learning Based Reinforcement Learning.

representation, allowing it to selectively attend to specific features or patterns in the stock price data. By integrating LSTM layers and attention mechanisms, the agent can capture long-term dependencies and identify important information for decision-making. This technique enhances the agent's ability to learn and make informed trading decisions, taking into account temporal dynamics and focusing on salient features.

In this research, we propose a deep learning model architecture for stock market trading that combines LSTM (Long Short-Term Memory) networks with an attention mechanism. The model is designed to learn from historical stock price data and make predictions on future stock price movements. The architecture begins with an input layer, defined by the shape of the state data. In this case, the state data has a shape of (state_size, 1), indicating the number of time steps (state_size) and the number of features (1). The LSTM layer is then applied to the input layer with 64 units. This layer is set to return sequences, allowing the model to capture temporal dependencies in the stock price data. Another LSTM layer with 64 units and return_sequences set at True follows the previous LSTM layer. This layer further enhances the model's ability to capture complex patterns and temporal dynamics in the data. Next, an attention mechanism is incorporated into the model architecture. This mechanism aims to highlight important features or patterns in the LSTM output. The attention layer is created using the TimeDistributed Dense layer, which applies a dense transformation to each time step of the LSTM output. This is followed by flattening the output and passing it through a Dense layer with a softmax activation function. The softmax activation allows the attention weights to be normalized and interpreted as a probability distribution across the time steps. To ensure compatibility with the LSTM output, the attention weights are repeated using the RepeatVector layer, replicating the attention weights across the time steps. The Permute layer then reshapes the attention weights to match the LSTM output shape. The Multiply layer combines the LSTM output with the attention weights element-wise multiplying them together. This process amplifies the importance of certain time steps, as determined by the attention weights, while suppressing others. The resulting tensor is flattened and passed through two Dense layers with 32 and 8 units, respectively. These layers introduce non-linear transformations to further process the information learned from the LSTM and attention layers. Finally, the output layer consists of a Dense layer with self. action_size units, where self.action_size represents the number of possible actions the agent can take. The activation function for this layer is set to linear, allowing the model to output continuous values. The overall model, defined using the Keras Model class, takes the inputs and outputs as parameters. It is then compiled with the mean squared error (MSE) loss function and the Adam optimizer with a learning rate of 0.001. This configuration enables the model to learn and optimize its parameters based on the MSE between predicted and target values. By employing this model architecture, we aim to enhance the agent's ability to capture temporal dependencies, identify important features, and make informed decisions in stock market trading. The mathematical equations of Deep LSTM Q-Learning (DLQL) are as follows:

$$i_t = \sigma(W_i[X_t,\ H_{t-1}] + b_i) \tag{9}$$

$$f_i = \sigma\left(W_f[X_t, H_{t-1}] + b_f\right) \tag{10}$$

$$o_t = \sigma(W_o[X_t,\ H_{t\_1}] + b_o) \tag{11}$$

$$c_t = \tanh(W_c[X_t,\ H_{t-1}] + b_c) \tag{12}$$

$$h_t = o_t \tanh(c_t) \tag{13}$$

$$Q(S_t,\ a_t; h_t) = Q(S_t,\ a_t; h_t) + \rho(R_{t+1} + \gamma \max(Q(S_{t+1}, a_t; h_t) - Q(S_t, a_t; h_t)) \tag{14}$$

Where S_t is the current state, a_t is the action taken in the current state, ρ is the learning rate determines the step size for updating Q-values, R_{t+1} is the reward received for taking action a_t in state S_t and transitioning for the next state S_{t+1}, γ is the discount factor that balances the importance of immediate rewards and future rewards, $\max(Q(S_{t+1}, a_t)$ is the maximum Q value over all possible actions a_t in the next state S_{t+1}, i_t is the input gate at time step t, f_i is the forget gate at step i, o_t is the output gate at step o, c_t is the cell state at step t, h_t is the LSTM hidden state at step t, σ is the sigmoid activation function, W_i, W_f, W_o, W_c are the weight matrices, b_i, b_f, b_o, b_c are the bias vector, X_t is the input sequence at time step t.

3.6.2. Deep LSTM Q-Learning (DLQL)

Deep LSTM Q-Learning leverages the power of LSTM networks in the Q-Learning framework. LSTM networks are capable of capturing and preserving long-term dependencies in sequential data, making them well-suited for modeling time-series stock price data. By utilizing LSTM layers, the agent can effectively capture the temporal patterns and dependencies in the stock market environment. The Deep LSTM Q-Learning approach enhances the agent's ability to learn and adapt to complex market dynamics, enabling it to make more accurate predictions and decisions.

In this study, we propose a deep learning model architecture for stock market trading using a Sequential model. The model combines LSTM (Long Short-Term Memory) networks with dense layers to learn from historical stock price data and make predictions on future stock price movements. The model begins with an LSTM layer, which takes as input a sequence of historical stock price data. The LSTM layer has 64 units, allowing it to capture complex patterns and temporal dependencies in the data. The input shape is defined as (self.state_size, 1), representing the number of time steps (self.state_size) and the number of features (1) for each time step. Following the LSTM layer, a dense layer is added with 32 units and a rectified linear activation function (ReLU). This layer introduces non-linear transformations to the output of the LSTM layer, enabling the model to learn complex relationships between the input data and the target variable. Another dense layer with 8 units and a ReLU activation function is added to further process the information learned from the previous layers. This additional layer enhances the model's capacity to capture higher-level features and patterns in the data. The final dense layer has self.action_size units, representing the number of possible actions the agent can take in the stock market trading environment. The activation function for this layer is set to linear, allowing the model to output continuous values, which is suitable for regression tasks. To train the model, the mean squared error (MSE) loss function is used, which measures the discrepancy between the predicted values and the target values. The Adam optimizer with a learning rate of 0.001 is employed to optimize the model's parameters and update the weights during training. By utilizing this model architecture, we aim to empower the agent with the ability to learn and make informed decisions based on historical stock price data. The LSTM layers capture temporal dependencies, while the dense layers introduce non-linear transformations and enable the model to approximate complex relationships in the data. The model is trained to minimize the MSE loss, optimizing its performance in predicting future stock price movements. Through the proposed architecture, we strive to enhance the agent's capability to analyze and interpret stock market data, leading to more accurate predictions and informed decision-making in stock market trading scenarios. The mathematical equation of Deep LSTM Attention Q-Learning (DLAQL) is as follows:

$$e_t = \tanh(W_a\left[H_t, H^k\right] \tag{15}$$

$$\widehat{\theta}_t = softmax\left(W_a^1 e_t\right) \tag{16}$$

$$\theta_t = \widehat{\theta}_t | \sum_k \widehat{\theta}_t^{\,k} \tag{17}$$

$$n_t = \sum_k \widehat{\theta}_t^{\,k} H^k \tag{18}$$

$$i_t = \sigma(W_i[n_t,\ H_{t-1}] + b_i) \tag{19}$$

$$f_i = \sigma\left(W_f[n_t, H_{t-1}] + b_f\right) \tag{20}$$

$$o_t = \sigma(W_o[n_t,\ H_{t\_1}] + b_o) \tag{21}$$

$$c_t = \tanh(W_c[n_t,\ H_{t-1}] + b_c) \tag{22}$$

$$h_t = o_t \tanh(c_t) \tag{23}$$

$$Q(S_t,\ a_t; h_t) = Q(S_t,\ a_t; h_t) + \rho(R_{t+1} + \gamma \max(Q(S_{t+1}, a_t; h_t) - Q(S_t, a_t; h_t)) \tag{24}$$

Where S_t is the current state, a_t is the action taken in the current state, ρ is the learning rate determines the step size for updating Q-values, R_{t+1} is the reward received for taking action a_t in state S_t and transitioning for the next state S_{t+1}, γ is the discount factor that balances the importance of immediate rewards and future rewards, $\max(Q(S_{t+1}, a_t)$ is the maximum Q value over all possible actions a_t in the next state S_{t+1}, i_t is the input gate at time step t, f_i is the forget gate at step i, o_t is the output gate at step o, c_t is the cell state at step t, h_t is the LSTM hidden state at step t, σ is the sigmoid activation function, W_i, W_f, W_o, W_c are the weight matrices, b_i, b_f, b_o, b_c are the bias vector, n_t is the attended input sequence at time step t, W_a, W_a^1 are the weight matrices, e_t is the attention weight.

4. Result and discussion

Fig. 2 is the historical stock price of CVE, MPLX, LNG and SU over the given period. The y-axis represents the stock price in the currency of the market being considered while the x-axis represents the timeline, with dates ranging from the earliest to the latest available data points. The line graph shows the movement of CVE, MPLX, LNG and SU stock prices over time, reflecting the fluctuations in its market value. LNG has the highest currency price over time, providing insights into its market performance.

Table 2 presents the descriptive statistics of the stock prices for CVE, MPLX, LNG, and SU. The table provides information on the mean, standard deviation (std), minimum (min), 25th percentile (25 %), median (50 %), 75th percentile (75 %), and maximum (max) values for each stock. The stock price of CVE has a mean value of 17.606423, with a standard deviation of 1.290118. The minimum price observed is 15.150000, while the 25th percentile, median, and 75th percentile prices are 16.505000, 17.540001, and 18.595000, respectively. The maximum price recorded for CVE is 20.400000. For LNG, the mean stock price is 149.497561, with a standard deviation of 5.021431. The minimum price is 137.750000, and the 25th, 50th, and 75th percentile prices are 146.579994, 149.410004, and 152.794998, respectively. The maximum price for LNG is 164.380005. MPLX exhibits a mean stock price of 34.245285, with a standard deviation of 0.545018. The minimum price observed is 32.580002, and the 25th, 50th, and 75th percentile prices are 33.855000, 34.250000, and 34.740002, respectively. The maximum price recorded for MPLX is 35.119999. Lastly, SU has a mean stock price of 31.228618, with a standard deviation of 2.064255. The minimum price is 28.000000, and the 25th, 50th, and 75th percentile prices are 29.285001, 30.840000, and 33.025000, respectively. The maximum price for SU is 35.299999.

Table 3 presents the Sharpe Ratio and Max Drawdown metrics for the stocks CVE, LNG, MPLX, and SU. The Sharpe Ratio is a measure of risk-adjusted return, indicating how well the returns of an investment compensate for its volatility. A higher Sharpe Ratio signifies better risk-adjusted performance. On the other hand, Max Drawdown measures the maximum loss experienced from a peak to a trough before a new peak is achieved. For CVE, the Sharpe Ratio is −1.1732, indicating a negative risk-adjusted return. This suggests that the investment's return does not compensate adequately for its volatility. The Max Drawdown for CVE is 0.2573, representing the maximum loss experienced during the investment period. LNG has a Sharpe Ratio of −1.5377, indicating a negative risk-adjusted return with higher volatility. The Max Drawdown for LNG is 0.1620, representing the maximum loss experienced before a new peak is reached. MPLX exhibits a significantly lower Sharpe Ratio of −4.3127, indicating a relatively poor risk-adjusted return compared to the other stocks. The Maximum Drawdown for MPLX is 0.0538, reflecting a relatively smaller loss experienced during the investment period. For SU, the Sharpe Ratio is −1.4714, suggesting a negative risk-adjusted return. The Maximum Drawdown for SU is 0.2067, representing the maximum loss experienced before a new peak is achieved. Therefore, based on the available information, CVE appears to have relatively better risk-adjusted performance compared to LNG, MPLX, and SU.

Table 4 presents the stock recommendations (BUY/SELL) for CVE, LNG, MPLX, and SU using the DLAQL, DLQL and LSTM models. The models provide recommendations based on their respective algorithms and criteria. For CVE, the DLAQL model recommends a BUY for 68 instances and a SELL for 66 instances. The DLQL model suggests a BUY for 52 instances and a SELL for 52 instances while the LSTM model suggests

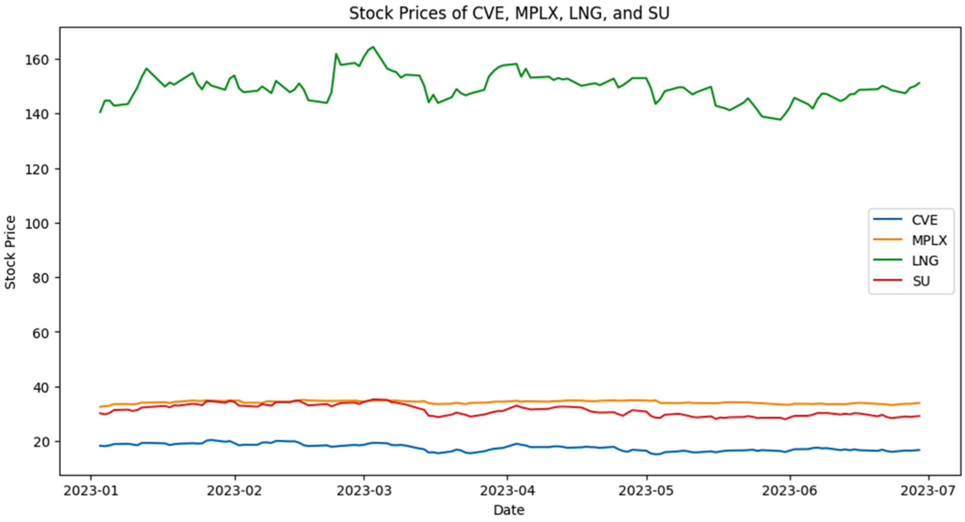

Fig. 2. Stock prices of CVE, MPLX, LNG and SU.

Table 2
Description of stock prices of CVE, MPLX, LNG and SU.

Stocks	Mean	Std	Min	25 %	50 %	75 %	max
CVE	17.606423	1.290118	15.150000	16.505000	17.540001	18.595000	20.400000
LNG	149.497561	5.021431	137.750000	146.579994	149.410004	152.794998	164.380005
MPLX	34.245285	0.545018	32.580002	33.855000	34.250000	34.740002	35.119999
SU	31.228618	2.064255	28.000000	29.285001	30.840000	33.025000	35.299999

Table 3
Risk-adjusted performance and maximum drawdown for CVE, LNG, MPLX, and SU.

Stocks	Sharpe ratio	Max drawdown
CVE	−1.1732	0.2573
LNG	−1.5377	0.1620
MPLX	−4.3127	0.0538
SU	−1.4714	0.2067

Table 4
Stock recommendations (BUY/SELL) for CVE, LNG, MPLX, and SU Using DLAQL and DLQL models.

Stocks	Model	Buy	Sell
CVE	DLAQL	68	66
	DLQL	52	52
	LSTM	50	50
LNG	DLAQL	59	59
	DLQL	56	56
	LSTM	54	54
MPLX	DLAQL	63	58
	DLQL	57	57
	LSTM	50	50
SU	DLAQL	66	66
	DLQL	59	59
	LSTM	56	56

a BUY for 50 instances and a SELL for 50 instances. Regarding LNG, the DLAQL model recommends a BUY for 59 instances and a SELL for 59 instances. The DLQL model indicates a BUY for 56 instances and a SELL for 56 instances while LSTM model suggests a BUY for 54 instances and a SELL for 54 instances. For MPLX, the DLAQL model suggests a BUY for 63 instances and a SELL for 58 instances. The DLQL model provides a BUY recommendation for 57 instances and a SELL recommendation for 57 instances while LSTM model suggests a BUY for 50 instances and a SELL for 50 instances. Regarding SU, the DLAQL model recommends a BUY for 66 instances and a SELL for 66 instances. The DLQL model suggests a BUY for 59 instances and a SELL for 59 instances while LSTM model suggests a BUY for 56 instances and a SELL for 56 instances.

Table 5 presents the total profit and cumulative reward achieved for CVE, LNG, MPLX, and SU using the DLAQL, DLQL and LSTM models. The total profit represents the net gain or loss resulting from the trading decisions made by the respective models. Cumulative reward reflects the cumulative performance over the trading period. For CVE, the DLAQL model achieves a total profit of 5.23 units and a cumulative reward of 498.27, while the DLQL model achieves a total profit of 4.47 units and a lower cumulative reward of 169.22. The LSTM model achieves a total profit of 2.11 units and a lower cumulative reward of 150.52. For LNG, both models experience losses, but the DLAQL model incurs a smaller loss with a total profit of −11.11 units and a higher cumulative reward of 4336.89 compared to the DLQL model, which records a total profit of −16.39 units and a slightly lower cumulative reward of 4070.11 while LSTM incur a total loss of −19.43 units and a cumulative reward of 4445.21. For MPLX, the DLQL model achieves a total profit of 0.86 units and a higher cumulative reward of 451.76, while the DLAQL model incurs a total loss of −2.86 units and a lower cumulative reward of 147.97. However, the LSTM model incurs a total loss of −3.45 units with a cumulative reward of 205.61. For SU, the DLQL model outperforms the DLAQL model, generating a total profit of 2.65 units and a higher cumulative reward of 775.43, while the DLAQL model incurs a total loss of −4.11 units and a lower cumulative reward of 622.10. The LSTM model incurs a total loss of −5.24 units and a cumulative reward of 885.32. Based on this information, it can be observed that the DLQL model tends to perform better in terms of total profit and cumulative reward compared to the DLAQL model for the majority of the stocks.

Table 6 presents the average profit per episode and average win rate for CVE, LNG, MPLX, and SU using the DLAQL and DLQL models. The average profit per episode represents the average amount gained or lost in each trading episode, while the average win rate indicates the percentage of episodes in which a profit was achieved. For CVE, the DLAQL model has an average profit per episode of −0.15 units and an average win rate of 54.06 % while LSTM model has an average profit per episode of −4.30 units and an average win rate of 24.45 %. In comparison, the DLQL model records a lower average profit per episode of −4.99 units and a lower average win rate of 37.96 %. Regarding LNG, the DLAQL model achieves a higher average profit per episode of 6.35 units and a higher average win rate of 47.24 % while LSTM model has an average profit per episode of −0.42 units and an average win rate of 39.76 %. The DLQL model, on the other hand, reports a slightly negative average profit per episode of −0.35 units and a lower average win rate of 43.74 %. For MPLX, the DLAQL model has an average profit per episode of −1.96 units and an average win rate of 42.52 % while LSTM model has an average profit per episode of −2.43 units and an average win rate

Table 5
Total profit and cumulative reward for CVE, LNG, MPLX, and SU Using DLAQL and DLQL models.

Stocks	Model	Total profit	Cumulative reward
CVE	DLAQL	5.23	498.27
	DLQL	4.47	169.22
	LSTM	2.11	150.52
LNG	DLAQL	−11.11	4336.89
	DLQL	−16.39	4070.11
	LSTM	−19.43	4445.21
MPLX	DLAQL	−2.86	147.97
	DLQL	0.86	451.76
	LSTM	−3.45	205.61
SU	DLAQL	−4.11	622.10
	DLQL	2.65	775.430
	LSTM	−5.24	885.321

Table 6
Average profit per episode and average win rate for CVE, LNG, MPLX, and SU Using DLAQL and DLQL models.

Stocks	Model	Average profit per episode	Average win rate (%)
CVE	DLAQL	−0.15	54.06
	DLQL	−4.99	37.96
	LSTM	−4.30	24.45
LNG	DLAQL	6.35	47.24
	DLQL	−0.35	43.74
	LSTM	−0.42	39.76
MPLX	DLAQL	−1.96	42.52
	DLQL	0.89	51.65
	LSTM	−2.43	34.56
SU	DLAQL	−7.15	52.10
	DLQL	−4.30	52.17
	LSTM	−8.23	56.32

of 34.56 %. The DLQL model shows a slightly positive average profit per episode of 0.89 units and a higher average win rate of 51.65 %. In the case of SU, the DLAQL model records an average profit per episode of −7.15 units and an average win rate of 52.10 % while LSTM model has an average profit per episode of −8.23 units and an average win rate of 56.32 %. The DLQL model reports an average profit per episode of −4.30 units and a similar average win rate of 52.17 %. Based on the average profit per episode and average win rate across the stocks, the DLQL model generally demonstrates better performance compared to the DLAQL model.

Fig. 3 depicts the loss of CVE in the DLAQL model. The x-axis represents the episode numbers, while the y-axis represents the loss values. Notably, the figure reveals a significant spike in loss with a value of 10 between the 3000 and 4000-episode range. This spike indicates a substantial increase in loss during this period, highlighting a potentially critical event or market condition affecting the performance of CVE in the DLAQL model. Furthermore, the figure shows the occurrence of the second-highest spikes, with loss values reaching 7 around the 5500-episode mark. These spikes indicate another notable period of increased loss, potentially indicating a distinct event or market behavior impacting CVE in the DLAQL model. The observed spikes in loss provide valuable insights into the volatility and performance fluctuations of CVE within the DLAQL model.

Fig. 4 illustrates the loss of CVE in the DLQL. The x-axis represents the episode numbers, while the y-axis represents the corresponding loss values. Each bar in the figure represents the loss value for a specific episode. A notable observation in Fig. 4 is the presence of a spike with a value of 1.0 between the 5000 and 5500-episode range. This spike indicates a substantial increase in loss during this period, suggesting a significant event or market behavior affecting the performance of CVE in the DLQL model. Additionally, Fig. 4 displays another prominent spike with a value of 1.0 between the 6000 and 7000 episodes. This spike signifies another noteworthy period of elevated loss, potentially indicating a distinct event or market condition impacting the performance of CVE within the DLQL model. These observed spikes in loss shed light on the volatility and fluctuation patterns of CVE within the DLQL model.

Fig. 5 showcases the loss of LNG in the DLAQL. The x-axis represents the episode numbers, while the y-axis represents the corresponding loss values. Each bar in the figure represents the loss value for a specific episode. A prominent feature observed in Fig. 5 is the highest spike in loss, reaching a value of 700 around the 3500-episode mark. This spike indicates a substantial increase in loss during this specific episode, suggesting a significant event or market behavior that significantly impacted the performance of LNG in the DLAQL model. The occurrence of such a pronounced spike highlights the presence of a critical event or extreme market condition that resulted in a considerable loss for LNG within the DLAQL model during the specified episode.

Fig. 6 presents the loss of LNG in the DLQL. The x-axis represents the episode numbers, while the y-axis represents the corresponding loss values. Each bar in the figure represents the loss value for a specific episode. A significant observation in Fig. 6 is the presence of the highest spikes in loss, reaching a value of 300 around the 3000 and 4500-episode marks. These spikes indicate substantial increases in loss during these specific episodes, suggesting notable events or market behaviors that significantly impacted the performance of LNG in the DLQL model.

Fig. 7 showcases the loss of MPLX in the DLAQL. The x-axis represents the episode numbers, while the y-axis represents the corresponding loss values. Each bar in the figure represents the loss value for a specific episode. A notable feature observed in Fig. 7 is the presence of the highest spikes in loss, reaching a value of 1.0 around the 3500-episode mark. This spike indicates a significant increase in loss during this specific episode, highlighting a potentially critical event or market behavior that substantially impacted the performance of MPLX in the DLAQL model. Additionally, Fig. 7 displays the occurrence of the second-highest spikes, with loss values reaching 0.7 around the 2800-episode mark. These spikes represent another notable period of increased loss, potentially indicating a distinct event or market condition that affected the performance of MPLX within the DLAQL model.

Fig. 8 depicts the loss of MPLX in the DLQL. A notable observation in Fig. 8 is the occurrence of the highest spikes in loss, reaching a value of 0.4 within the range of 1200 to 6200 episodes. These spikes indicate substantial increases in loss during this specific episode range, suggesting significant events or market behaviors that significantly impacted the performance of MPLX in the DLQL model. The presence of these pronounced spikes highlights critical episodes characterized by heightened losses for MPLX within the DLQL model.

Fig. 9 illustrates the loss of SU in the DLAQL. A notable feature observed in Fig. 9 is the occurrence of the highest spikes in loss, reaching a value of 8 around the 3800-episode mark. This spike indicates a significant increase in loss during this specific episode, highlighting a potentially critical event or market behavior that significantly impacted the performance of SU in the DLAQL model. Furthermore, Fig. 9 reveals another prominent spike with a value of 7 around the 5000-episode mark. This spike signifies another notable period of increased loss, potentially indicating a distinct event or market condition that affected the performance of SU within the DLAQL model. These observed spikes in loss provide valuable insights into the volatility and fluctuations in the performance of SU within the DLAQL model.

Fig. 10 showcases the loss of SU in the DLQL. A significant

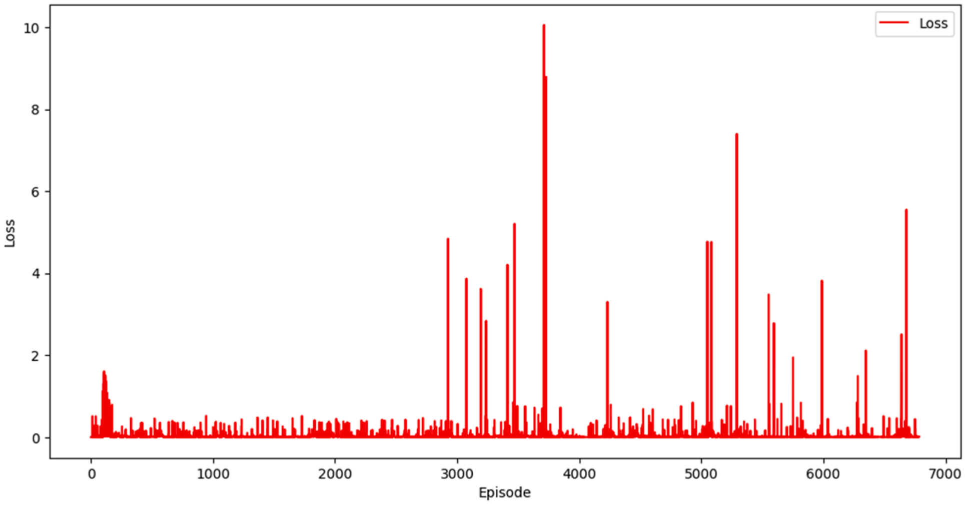

Fig. 3. Loss of CVE in DLAQL.

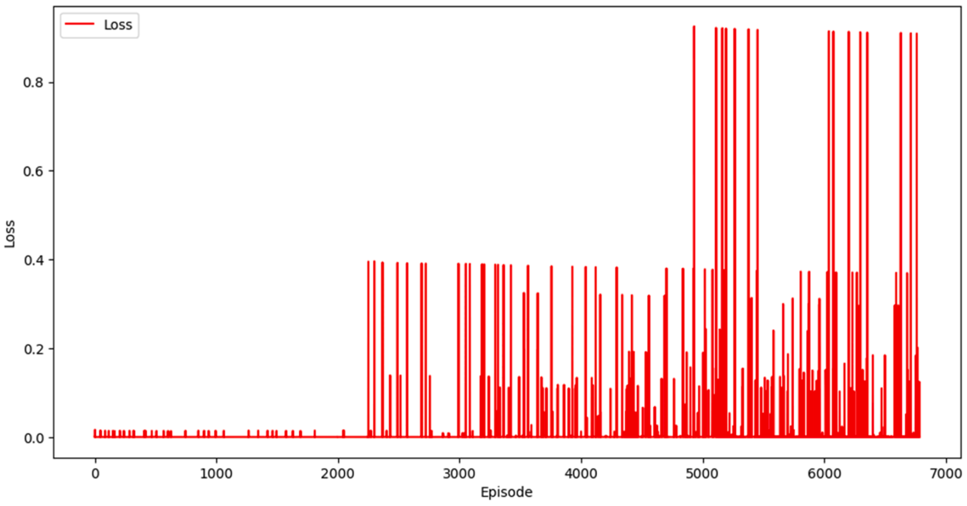

Fig. 4. Loss of CVE in DLQL.

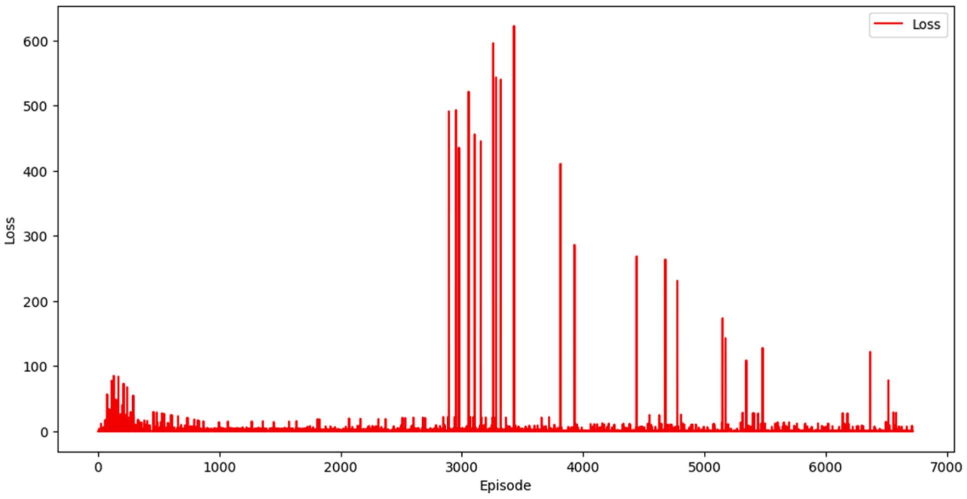

Fig. 5. Loss of LNG in DLAQL.

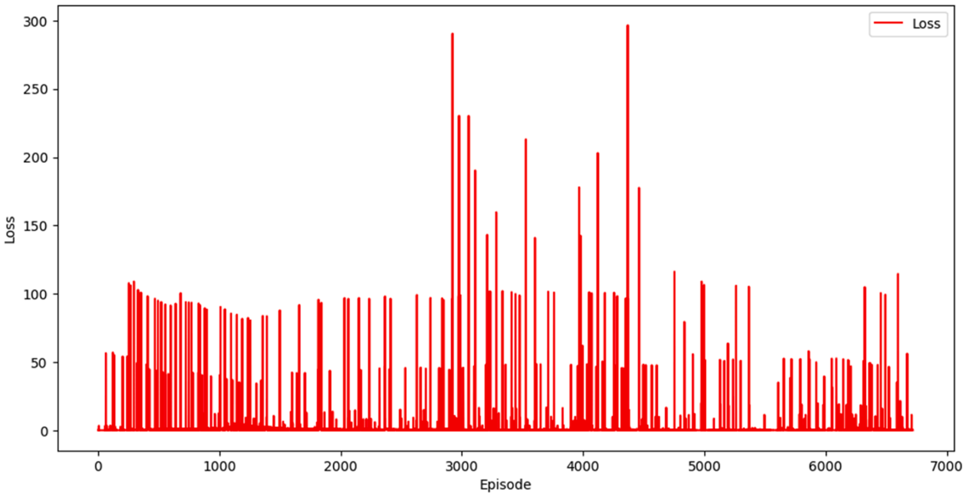

Fig. 6. Loss of LNG in DLQL.

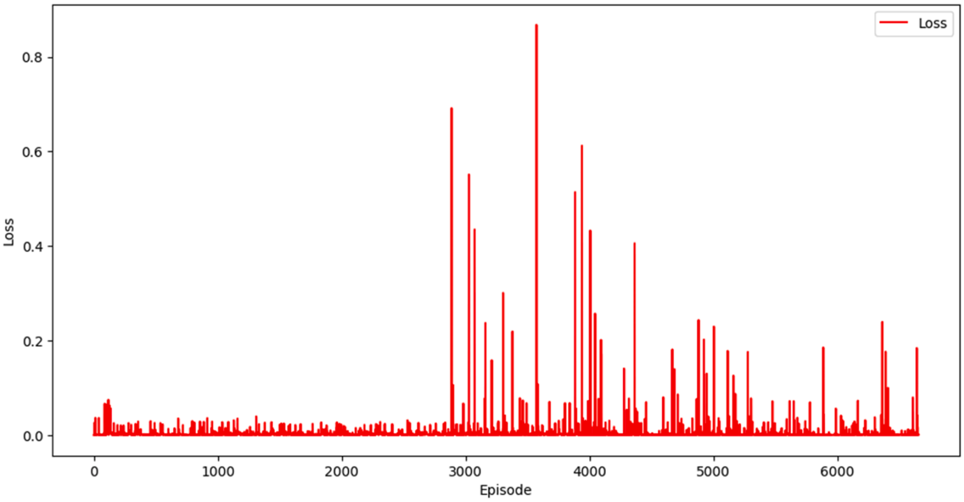

Fig. 7. Loss of MPLX in DLAQL.

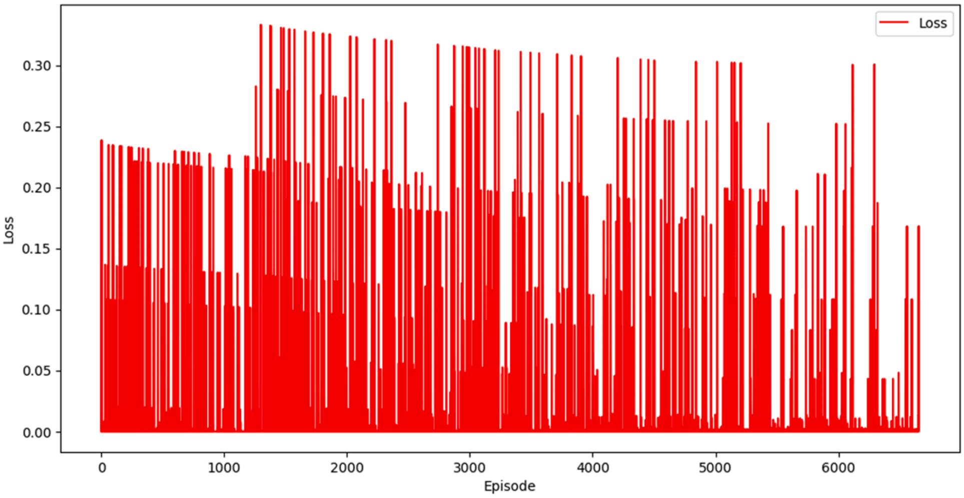

Fig. 8. Loss of MPLX in DLQL.

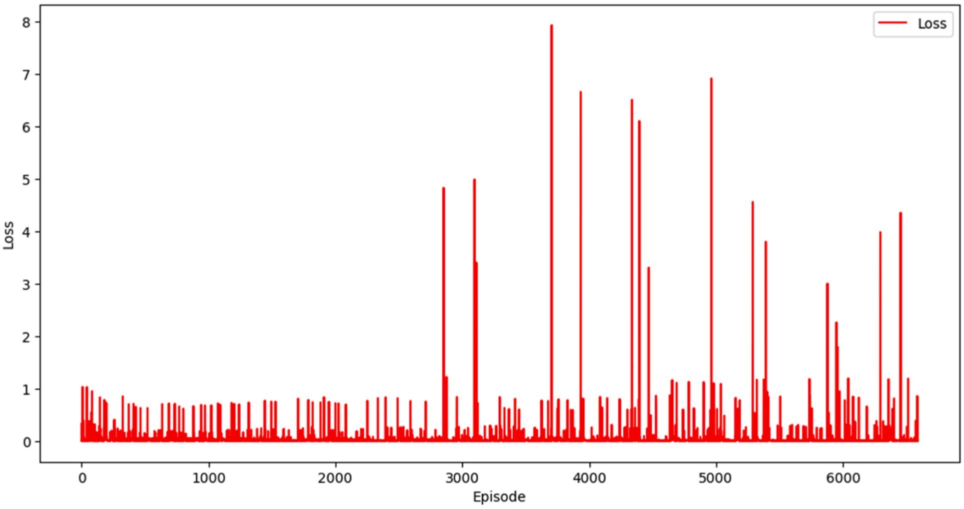

Fig. 9. Loss of SU in DLAQL.

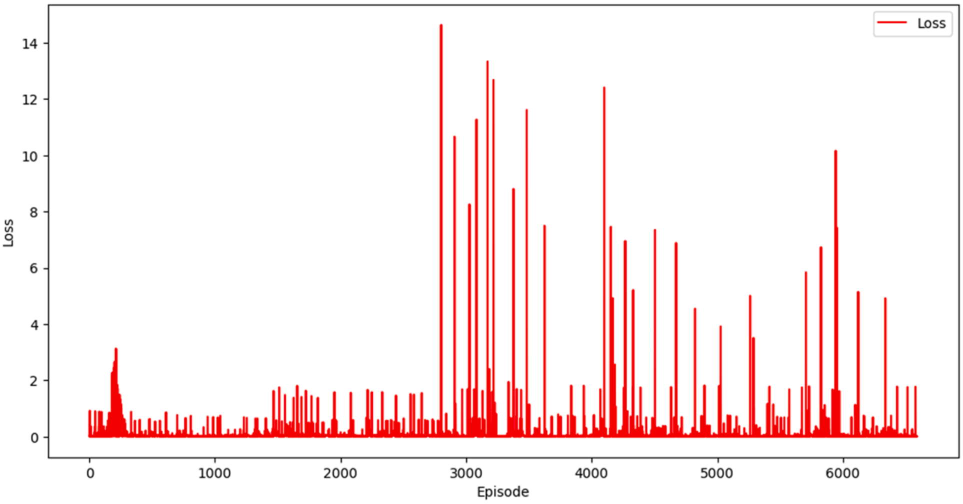

Fig. 10. Loss of SU in DLQL.

observation in Fig. 10 is the presence of the highest spike in loss, reaching a value of 15 at the 3800-episode mark. This spike indicates a substantial increase in loss during this specific episode, suggesting a notable event or market behavior that significantly impacted the performance of SU in the DLQL model. Additionally, Fig. 10 reveals another notable spike in loss at the 3100-episode mark. Although not as high as the highest spike, this spike signifies another significant period of increased loss, potentially indicating a distinct event or market condition that affected the performance of SU within the DLQL model.

Fig. 11 presents the buy and sell actions of CVE in the DLAQL. The x-axis represents the stock prices, while the y-axis represents the number of days for each buy and sell action. Each peak in the figure represents a specific stock price level. A significant observation in Fig. 11 is the presence of the highest peak at 22.3 stock prices. At this price level, it shows a sell action occurring in 20 days, indicating a decision to sell the stock when it reaches this peak price. Conversely, Fig. 11 also reveals the presence of the lowest peak at 15.1 stock prices. At this price level, it shows a selling action occurring in 84 days, suggesting a decision to sell the stock when it reaches this lowest price point. These observed buy and sell actions provide valuable insights into the trading decisions made for CVE within the DLAQL model. The presence of sell actions at the highest peak and the lowest peak indicates potential strategies to take advantage of price fluctuations and optimize investment returns.

Fig. 12 displays the buy and sell actions of CVE in the DLQL. A notable observation in the figure is the occurrence of a buy action at the highest peak of 22.3 stock prices, taking place in 20 days. This indicates a decision to purchase the stock when it reaches this peak price level within the DLQL model. In contrast, Fig. 12 also reveals a sell action at the lowest peak of 15.1 stock prices, occurring in 84 days. This suggests a decision to sell the stock when it reaches this lowest price point.

Fig. 13 presents the buy and sell actions of LNG in the DLAQL. The x-axis represents the stock prices, while the y-axis represents the number of days for each buy and sell action. Each peak in the figure corresponds to a specific stock price level. An important observation in Fig. 13 is the occurrence of a buy action at the highest peak of 165 stock prices, taking place over 40 days. This indicates a decision to purchase the stock when it reaches this peak price level within the DLAQL model. Conversely, Fig. 13 shows no action at the lowest peak of 110 stock prices for 100 days. This suggests that no buy or sell action was taken during this period when the stock price was at its lowest level. The presence of buy actions at the highest peak and the absence of any action at the lowest peak provide insights into the trading decisions made for LNG within the DLAQL model. It suggests a strategy of purchasing the stock when it reaches the highest price point but not taking any action when it reaches the lowest price level.

Fig. 14 illustrates the buy and sell actions of LNG in the DLQL. An important observation in Fig. 14 is the absence of any buy or sell action at the highest peak of 165 stock prices for 40 days. This indicates that no

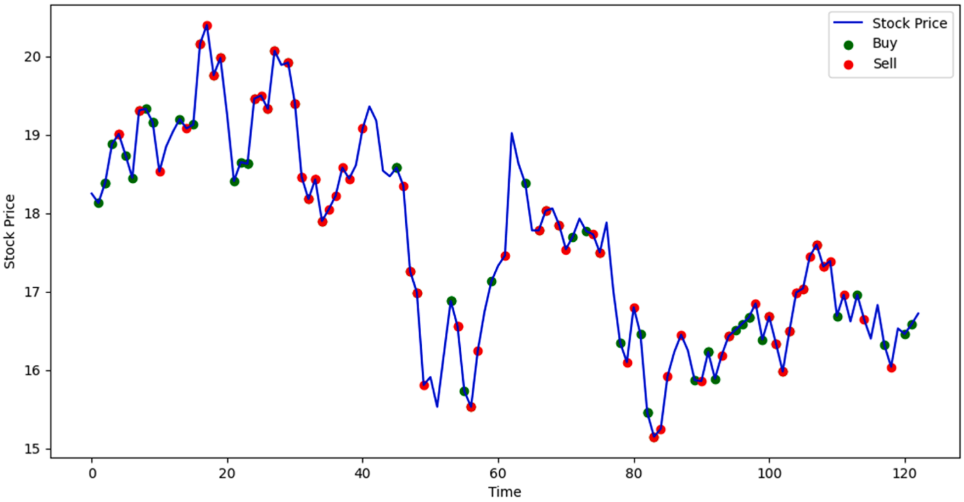

Fig. 11. Buy and sell actions of CVE in DLAQL.

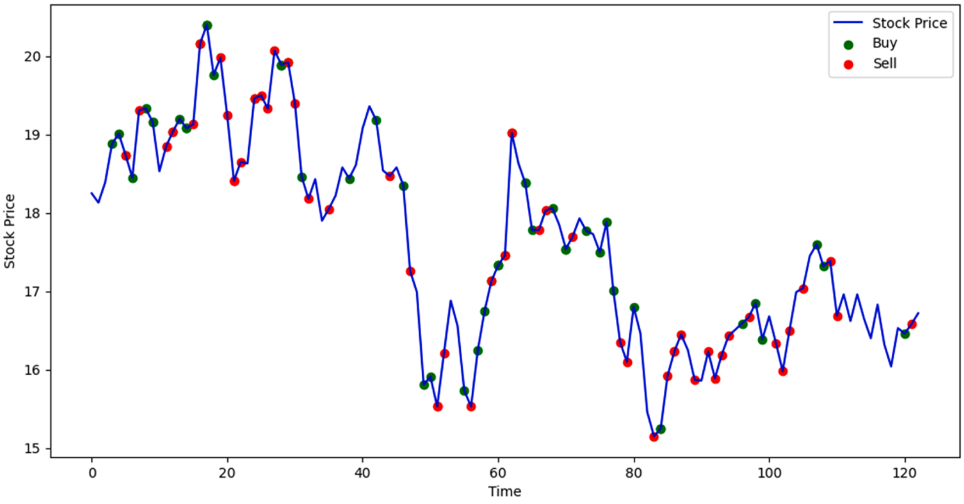

Fig. 12. Buy and sell actions of CVE in DLQL.

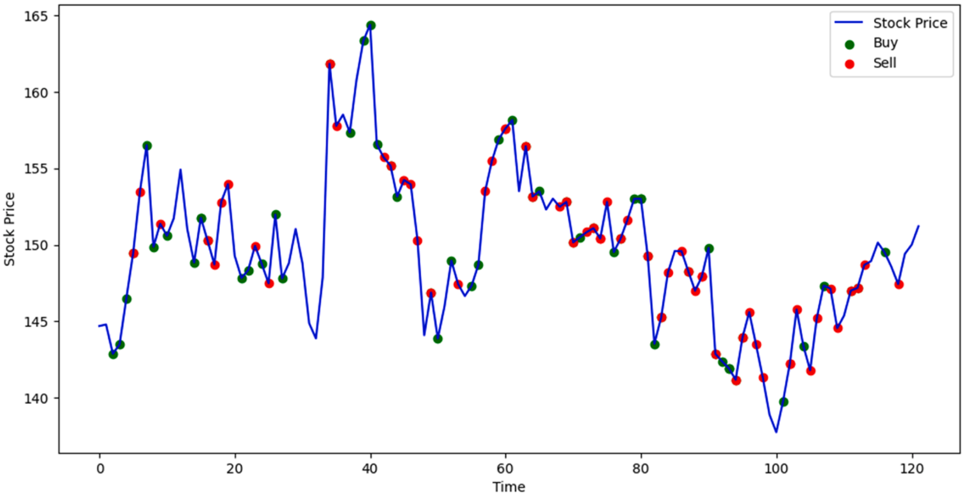

Fig. 13. Buy and sell actions of LNG in DLAQL.

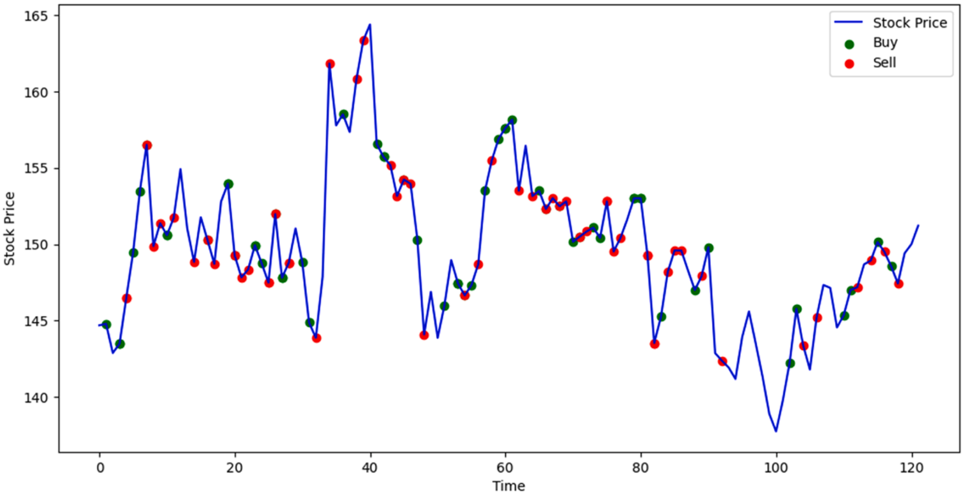

Fig. 14. Buy and sell actions of LNG in DLQL.

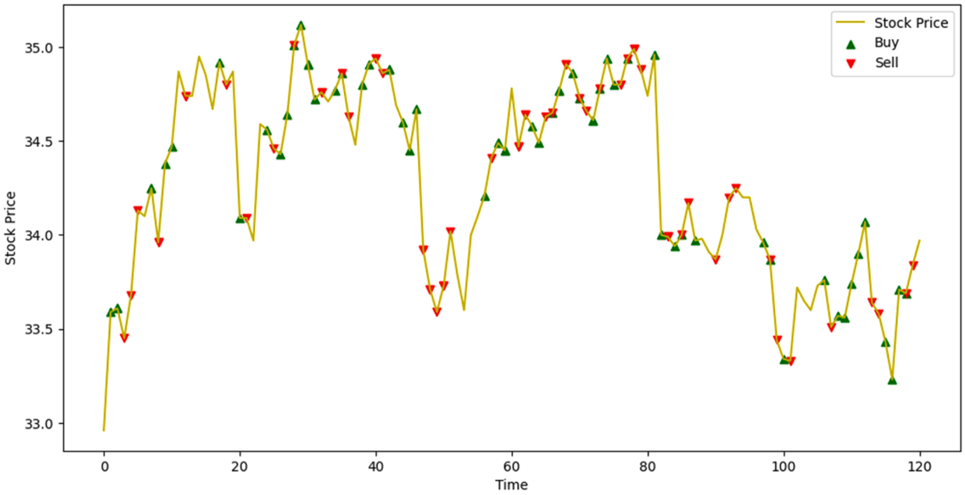

Fig. 15. Buy and sell actions of MPLX in DLAQL.

trading decision was made during this period when the stock price reached its highest level within the DLQL model. Similarly, Fig. 14 also shows no action at the lowest peak of 110 stock prices for 100 days. This suggests that no buy or sell action was taken during this period when the stock price was at its lowest level. The absence of buy and sell actions at both the highest and lowest price peaks provides insights into the trading decisions made for LNG within the DLQL model. It indicates a strategy of not taking any action at extreme price levels, potentially indicating a cautious or conservative approach to trading.

Fig. 15 showcases the buy and sell actions of MPLX in the DLAQL. The x-axis represents the stock prices, while the y-axis represents the number of days for each buy and sell action. Each peak in the figure corresponds to a specific stock price level. An interesting observation in Fig. 15 is the presence of buy actions at the highest peak of 35 stock prices, occurring over 35 days. This indicates a decision to purchase the stock when it reaches this highest price level within the DLAQL model. Furthermore, Fig. 15 reveals a buy action at the lowest peak of 34.2 stock prices, taking place over 118 days. This suggests a decision to buy the stock when it reaches this lowest price point. The occurrence of buy actions at both the highest and lowest price peaks provides insights into the trading decisions made for MPLX within the DLAQL model. It suggests a strategy of purchasing the stock when it reaches extreme price levels, potentially aiming to benefit from potential price reversals or take advantage of perceived value opportunities.

Fig. 16 depicts the buy and sell actions of MPLX in the DLQL. Remarkably, Fig. 16 demonstrates consistent buy actions at both the highest peak of 35 stock prices, occurring over 35 days, and the lowest peak of 34.2 stock prices, spanning 118 days. This indicates a consistent decision to purchase the stock when it reaches both the highest and lowest price levels within the DLQL model. The consistent buy actions at the highest and lowest price peaks provide valuable insights into the trading decisions made for MPLX within the DLQL model. It suggests a strategy of capitalizing on extreme price levels, potentially seeking opportunities for growth or perceived value in the stock.

Fig. 17 illustrates the buy and sell actions of SU in the DLAQL. The x-axis represents the stock prices, while the y-axis represents the number of days for each buy and sell action. Each peak in the figure corresponds to a specific stock price level. An interesting observation in Fig. 17 is the absence of any buy or sell action at the highest peak of 36 stock prices for 40 days. This indicates that no trading decision was made during this period when the stock price reached its highest level within the DLAQL model. Conversely, Fig. 17 reveals both a buy and sell action at the lowest peak of 28 stock prices, occurring over 100 days. This suggests that the model made a decision to buy and subsequently sell the stock when it reached this lowest price point.

Fig. 18 showcases the buy and sell actions of SU in the DLQL. An interesting observation in Fig. 18 is the presence of both buy and sell actions at the highest peak of 36 stock prices, occurring over 40 days.

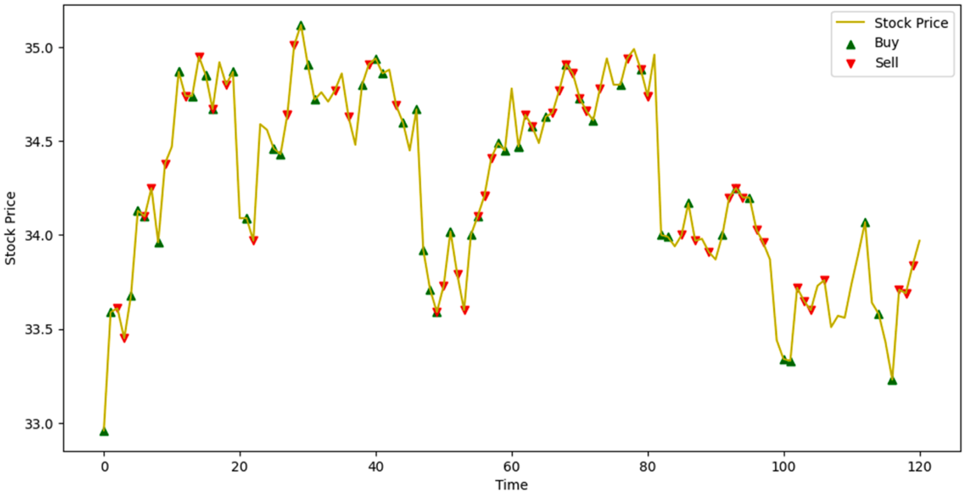

Fig. 16. Buy and sell actions of MPLX in DLQL.

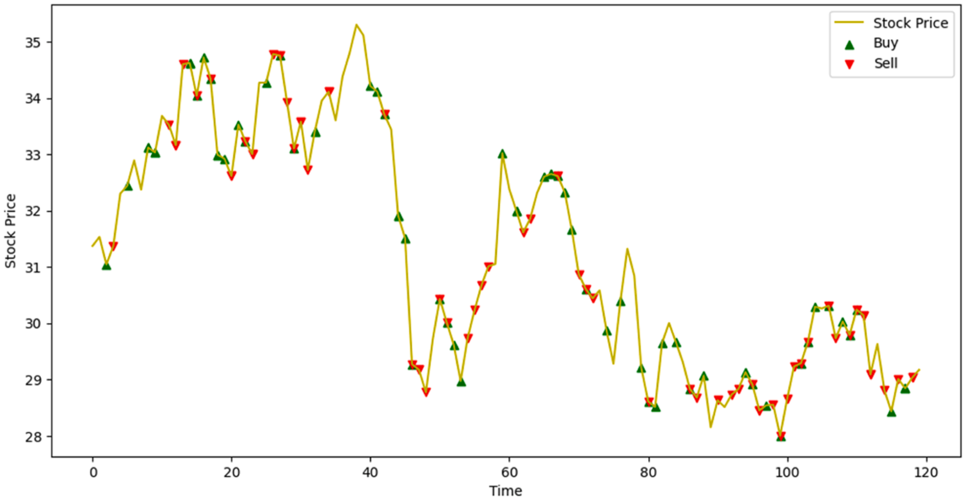

Fig. 17. Buy and sell actions of SU in DLAQL.

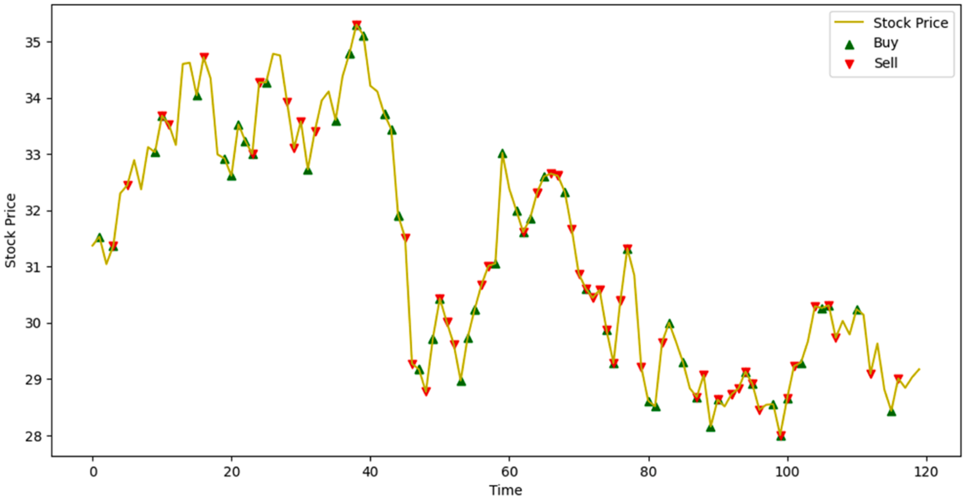

Fig. 18. Buy and sell actions of SU in DLQL.

This indicates a decision to both purchase and subsequently sell the stock when it reaches this highest price level within the DLQL model. Additionally, Fig. 18 reveals both a buy and sell action at the lowest peak of 28 stock prices, taking place over 100 days. This suggests a decision to buy the stock when it reaches this lowest price point and subsequently sell it at a later stage.

5. Conclusion

This study investigated the application of Deep LSTM Q-Learning (DLQL) and Deep LSTM-Attention Q-Learning (DLAQL) based reinforcement learning in the prediction of the oil and gas sector. The primary aim was to assess the performance and effectiveness of these models in forecasting key indicators and trends in the oil and gas industry. The motivation behind this study stemmed from the importance of accurate predictions in the oil and gas sector, which is characterized by complex market dynamics and various factors influencing price fluctuations. Accurate predictions can provide valuable insights for decision-making, risk management, and investment strategies in this sector. The findings of this study demonstrated the potential of DLQL and DLAQL models in predicting key indicators and trends in the oil and gas sector. Both models exhibited promising performance in capturing the temporal dependencies and complex patterns in the data, enabling accurate predictions of important variables such as oil and gas prices, production levels, and market trends. The significance of this study lies in its contribution to the existing body of knowledge in the field of oil and gas sector prediction. By employing DLQL and DLAQL models, this research expands the range of techniques available for forecasting in this sector and provides insights into their effectiveness. The conclusions drawn from this research indicate that DLQL and DLAQL models can be valuable tools for decision-makers and analysts in the oil and gas industry. The models' ability to leverage deep learning techniques and attention mechanisms allows for an improved understanding of complex relationships and patterns in the data, leading to more accurate predictions and informed decision-making. Future research in this area could focus on further refining and enhancing the DLQL and DLAQL models by incorporating additional features, exploring alternative architectures, or integrating other advanced techniques from the field of deep reinforcement learning. Future research can also focus on investigating the potential benefits of combining LSTM-based models with other deep learning architectures, such as convolutional neural networks (CNN) for better feature extraction or transformer models for capturing longer-range dependencies in time series data. Additionally, expanding the scope of the study to include other sectors or asset classes within the energy industry would provide a broader understanding of the models' performance and applicability. The adoption of deep learning models, such as LSTM and LSTM-Attention, comes with inherent complexity and significant computational demands. Implementing and training these models may necessitate substantial computational resources.

CRediT authorship contribution statement

David Opeoluwa Oyewola: Writing – review & editing, Visualization, Methodology, Investigation, Data curation. **Sulaiman Awwal Akinwunmi: . Temidayo Oluwatosin Omotehinwa:** .

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] D.O. Oyewola, E.G. Gbenga Dada, O.E. Olaoluwa, K. Al-Mustapha, Predicting Nigerian stock returns using technical analysis and machine learning, Eur. J. Electr. Eng. Comput. Sci. 3 (Mar. 2019) (2019) 2, https://doi.org/10.24018/ejece.2019.3.2.65.

[2] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, Deep learning for stock market prediction, Entropy (2020) mdpi.com, https://www.mdpi.com/1099-4300/22/8/840.

[3] D.P. Gandhmal, K. Kumar, Systematic analysis and review of stock market prediction techniques, Comput. Sci. Rev. (2019). https://www.sciencedirect.com/science/article/pii/S157401371930084X.

[4] W. Jiang, Applications of deep learning in stock market prediction: recent progress, Expert Syst. Appl. (2021). https://www.sciencedirect.com/science/article/pii/S0957417421009441.

[5] E. Hoseinzade, S. Haratizadeh, CNNpred: CNN-based stock market prediction using a diverse set of variables, Expert Syst. Appl. (2019). https://www.sciencedirect.com/science/article/pii/S0957417419301915.

[6] K. Zhang, G. Zhong, J. Dong, S. Wang, Y. Wang, Stock market prediction based on generative adversarial network, Procedia Comput. Sci. (2019). https://www.sciencedirect.com/science/article/pii/S1877050919302789.

[7] A. Thakkar, K. Chaudhari, Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions, Inform. Fus. (2021). https://www.sciencedirect.com/science/article/pii/S1566253520303481.

[8] A. Moghar, M. Hamiche, Stock market prediction using LSTM recurrent neural network, Procedia Comput. Sci. (2020). https://www.sciencedirect.com/science/article/pii/S1877050920304865.

[9] H. Chung, K. Shin, Genetic algorithm-optimized long short-term memory network for stock market prediction, Sustainability (2018) mdpi.com, https://www.mdpi.com/2071-1050/10/10/3765.

[10] A. Kelotra, P. Pandey, Stock market prediction using optimized deep-convlstm model, Big Data (2020) liebertpub.com, https://www.liebertpub.com/doi/abs/10.1089/big.2018.0143.

[11] H. Chung, K. Shin, Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction, Neural. Comput. Appl. (2020). https://link.springer.com/article/10.1007/s00521-019-04236-3.

[12] S.R. Das, D. Mishra, M. Rout, Stock market prediction using Firefly algorithm with evolutionary framework optimized feature reduction for OSELM method, Expert Syst. Applic.: X (2019). https://www.sciencedirect.com/science/article/pii/S2590188519300162.

[13] D.O. Oyewola, A. Ibrahim, J.A. Kwanamu, E.G. Dada, A new auditory algorithm in stock market prediction on oil and gas sector in Nigerian stock exchange, Soft Comput. Lett. (2021). https://www.sciencedirect.com/science/article/pii/S2666222121000034.

[14] D. Kumar, P.K. Sarangi, R. Verma, A systematic review of stock market prediction using machine learning and statistical techniques, in: Materials Today: Proceedings, Elsevier, 2022. https://www.sciencedirect.com/science/article/pii/S2214785320390337.

[15] S. Mokhtari, K.K. Yen, J. Liu, Effectiveness of Artificial Intelligence in Stock Market Prediction Based on Machine Learning, 2021 arXiv preprint *arXiv:2107.01031*, arxiv.org, https://arxiv.org/abs/2107.01031.

[16] G. Singh, Machine Learning Models in Stock Market Prediction, 2022 arXiv preprint *arXiv:2202.09359*, arxiv.org, https://arxiv.org/abs/2202.09359.

[17] Z. Fathali, Z. Kodia, L.B. Said, Stock market prediction of Nifty 50 index applying machine learning techniques, Appl. Artifi. Intell. (2022). https://www.tandfonline.com/doi/abs/10.1080/08839514.2022.2111134.

[18] P. Koukaras, C. Nousi, C. Tjortjis, Stock market prediction using microblogging sentiment analysis and machine learning, Telecom (2022) mdpi.com, https://www.mdpi.com/2673-4001/3/2/19.

[19] H. Kim, S. Jun, K.S. Moon, Stock market prediction based on adaptive training algorithm in machine learning, Quant. Finance (2022). https://www.tandfonline.com/doi/abs/10.1080/14697688.2022.2041208.

[20] M. Bansal, A. Goyal, A. Choudhary, Stock market prediction with high accuracy using machine learning techniques, Procedia Comput. Sci. (2022). https://www.sciencedirect.com/science/article/pii/S1877050922020993.

[21] K. Kumar, M.T.U. Haider, Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN–LSTM network, New Gener. Comput. (2021). https://link.springer.com/article/10.1007/s00354-020-00104-0.

[22] E. Hoseinzade, S. Haratizadeh, CNNPred: CNN-based Stock Market Prediction Using Several Data Sources, 2018 arXiv preprint *arXiv:1810.08923*, arxiv.org, https://arxiv.org/abs/1810.08923.

[23] J. Eapen, D. Bein, A. Verma, Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction, in: 2019 IEEE 9th Annual Computing, 2019 ieeexplore.ieee.org, https://ieeexplore.ieee.org/abstract/document/8666592/.

[24] A. Sharma, P. Tiwari, A. Gupta, P. Garg, Use of LSTM and ARIMAX algorithms to analyze impact of sentiment analysis in stock market prediction, Intell. Data Commun. (2021). https://link.springer.com/chapter/10.1007/978-981-15-9509-7_32.

[25] M. Ali, D.M. Khan, H.M. Alshanbari, A.A.H. El-Bagoury, Prediction of complex stock market data using an improved hybrid emd-lstm model, Appl. Sci. (2023) mdpi.com, https://www.mdpi.com/2076-3417/13/3/1429.

[26] K. Srijiranon, Y. Lertratanakham, T. Tanantong, A hybrid Framework Using PCA, EMD and LSTM methods for stock market price prediction with sentiment analysis, Appl. Sci. (2022) mdpi.com, https://www.mdpi.com/2076-3417/12/21/10823.

[27] E. Koo, G. Kim, A hybrid prediction model integrating garch models with a distribution manipulation strategy based on lstm networks for stock market volatility, IEEE Access (2022) ieeexplore.ieee.org, https://ieeexplore.ieee.org/abstract/document/9745535/.

[28] A. Ifleh, M El Kabbouri, Moroccan stock market prediction using LSTM model on a daily data, in: Intelligent Systems: Proceedings of SCIS 2021, Springer, 2021. https://link.springer.com/chapter/10.1007/978-981-16-2248-9_2.

[29] C. Bergström, O. Hjelm, Impact of Time Steps On Stock Market Prediction with LSTM, 2019 diva-portal.org, https://www.diva-portal.org/smash/record.jsf?pid=diva2:1361305.

[30] V. Kuber, D. Yadav, A.K. Yadav, Univariate and Multivariate LSTM Model for Short-Term Stock Market Prediction, 2022 arXiv preprint *arXiv:2205.06673*, arxiv.org, https://arxiv.org/abs/2205.06673.

[31] D.O. Oyewola, L.A. Oladimeji, S.A. Julius, L.B. Kachalla, E.G. Dada, Optimizing sentiment analysis of Nigerian 2023 presidential election using two-stage residual long short term memory, Heliyon 9 (4) (2023) e14836, https://doi.org/10.1016/j.heliyon.2023.e14836, 2023.

[32] A.B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech Recognition Using Deep Neural Networks: a Systematic Review, IEEE Access 7 (2019) 19143–19165, https://doi.org/10.1109/ACCESS.2019.2896880, 2019.

[33] P. Ladosz, L. Weng, M. Kim, H. Oh, Exploration in deep reinforcement learning: a survey, Inform. Fus. (2022). https://www.sciencedirect.com/science/article/pii/S1566253522000288.

[34] M. Sewak, Deep Reinforcement Learning, Springer, 2019. https://link.springer.com/content/pdf/10.1007/978-981-13-8285-7.pdf.

[35] H. Dong, H. Dong, Z. Ding, S. Zhang, Chang, Deep Reinforcement Learning, Springer, 2020. https://link.springer.com/content/pdf/10.1007/978-981-15-4095-0.pdf.

[36] A. Heuillet, F. Couthouis, N. Díaz-Rodríguez, Explainability in deep reinforcement learning, Knowl.-Based Syst. (2021). https://www.sciencedirect.com/science/article/pii/S0950705120308145.

[37] Z. Zhou, S. Kearnes, L. Li, R.N. Zare, P. Riley, Optimization of molecules via deep reinforcement learning, Sci. Rep. (2019) nature.com, https://www.nature.com/articles/s41598-019-47148-x.

[38] A. Stooke, P. Abbeel, rlpyt: A research Code Base for Deep Reinforcement Learning in Pytorch, 2019 arXiv preprint *arXiv:1909.01500*, arxiv.org, https://arxiv.org/abs/1909.01500.

[39] S.K. Zhou, H.N. Le, K. Luu, H.V. Nguyen, N. Ayache, Deep reinforcement learning in medical imaging: a literature review, Med. Image Anal. (2021). https://www.sciencedirect.com/science/article/pii/S1361841521002395.

[40] A. Stooke, P. Abbeel, Accelerated Methods for Deep Reinforcement Learning, 2018 arXiv preprint *arXiv:1803.02811*, arxiv.org, https://arxiv.org/abs/1803.02811.

[41] S.K. Lakshminarayanan, J.P. McCrae, A comparative study of SVM and LSTM deep learning algorithms for stock market prediction, AICS (2019) ceur-ws.org, https://ceur-ws.org/Vol-2563/aics_41.pdf.

[42] S.K. Chandar, Fusion model of wavelet transform and adaptive neuro fuzzy inference system for stock market prediction, J. Amb. Intell. Human. (2019). https://link.springer.com/article/10.1007/s12652-019-01224-2.

[43] C. Zhang, N.N.A. Sjarif, R.B. Ibrahim, Decision fusion for stock market prediction: a systematic review, IEEE Access (2022) ieeexplore.ieee.org, https://ieeexplore.ieee.org/abstract/document/9847239/.

[44] S.S. Pal, S. Kar, Time series forecasting for stock market prediction through data discretization by fuzzistics and rule generation by rough set theory, Math. Comput. Simul (2019). https://www.sciencedirect.com/science/article/pii/S0378475419300011.

[45] A. Yadav, C.K. Jha, A. Sharan, Optimizing LSTM for time series prediction in Indian stock market, Procedia Computer Science (2020). https://www.sciencedirect.com/science/article/pii/S1877050920307237.

[46] Y. Baek, H.Y. Kim, ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module, Expert Syst. Appl. (2018). https://www.sciencedirect.com/science/article/pii/S0957417418304342.

[47] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, H. Fujita, Adaptive stock trading strategies with deep reinforcement learning methods, Inform. Sci. (2020). https://www.sciencedirect.com/science/article/pii/S0020025520304692.

[48] T. Théate, D. Ernst, An application of deep reinforcement learning to algorithmic trading, Expert Syst. Appl. (2021). https://www.sciencedirect.com/science/article/pii/S0957417421000737.

[49] Y. Li, P. Liu, Z. Wang, Stock trading strategies based on deep reinforcement learning, Sci. Program. (2022) hindawi.com, https://www.hindawi.com/journals/sp/2022/4698656/.

[50] T. Kabbani, E. Duman, Deep reinforcement learning approach for trading automation in the stock market, IEEE Access (2022) ieeexplore.ieee.org, https://ieeexplore.ieee.org/abstract/document/9877940/.

[51] A. Brim, N.S. Flann, Deep reinforcement learning stock market trading, utilizing a CNN with candlestick images, PLoS One (2022) journals.plos.org, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0263181.

[52] S. Bajpai, Application of Deep Reinforcement Learning For Indian stock Trading Automation, 2021 arXiv preprint *arXiv:2106.16088*, arxiv.org, https://arxiv.org/abs/2106.16088.

[53] Y. Li, W. Zheng, Z. Zheng, Deep robust reinforcement learning for practical algorithmic trading, IEEE Access (2019) ieeexplore.ieee.org, https://ieeexplore.ieee.org/abstract/document/8786132/.

[54] S.H. Kim, D.Y. Park, K.H. Lee, Hybrid deep reinforcement learning for pairs trading, Appl. Sci. (2022) mdpi.com, https://www.mdpi.com/2076-3417/12/3/944.

[55] X.Y. Liu, Z. Xiong, S. Zhong, H. Yang, A. Walid, Practical Deep Reinforcement Learning Approach for Stock Trading, 2018. https://arxiv.org/abs/1811.07522.

[56] A. Shavandi, M. Khedmati, A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets, Expert Syst. Appl. (2022). https://www.sciencedirect.com/science/article/pii/S0957417422013082.