Reinforcement Learning Models in Stock Trading

Efe Precious Onakpojeruo[1,2] [0000-0001-8582-409X], Berna Uzun [1,3] [0000-0002-5438-8608], Fadi Al-Turjman[4]

[1] Operational Research Center in Healthcare, Department of Biomedical Engineering, Near East University, Nicosia 99138, Mersin 10, Turkey.

[2] Department of Biomedical Engineering, Near East University, Nicosia 99138, Mersin 10, Turkey

[3] Department of Mathematics, Near East University, Nicosia 99138, Mersin 10, Turkey.

[4] Artificial Intelligence Engineering Department, Research Centre for AI and IoT, AI and Robotics Institute, Near East University, Mersin 10, Turkey.

Correspondence: E.P.O.: efeprecious.onakpojeruo@neu.edu.tr

Abstract: Many researchers and professional stock traders have struggled with the specialty of figuring out stock prices. The study area of stock value prediction has piqued financial experts' interest greatly. Many speculators are adept at predicting the stock market's future direction, which allows for decent and profitable speculation. Brokers, speculators, and professional traders can provide crucial information on the stock market's future direction with the use of tremendous and strong prediction frameworks for the stock market. In this study, Reinforcement Learning (RL) models are shown to have the best predictive and trading signal accuracy for the stock market. The Preference Ranking Organization Method for Enrichment Evaluation (fuzzy PROMETHEE) multicriteria decision-making (MCDM) method was used to evaluate the RL models developed in this study. the following; accuracy, precision, consistency in making profits, simplicity in implementation, profit optimization rate, volatility rate/speed, reliability, and speed were employed to evaluate the performance of the models. The results from this study showed that with a net flow of 0.0823, DDQN was determined as the most favorable and preferred RL model in stock trading. DQN, Dueling QN, and CNN came second, third, and fourth, with net flows of 0.0364, −0.0142, and −0.0465, respectively. RNN-LSTM with a net flow of −0.0581 was the least preferred alternative. The obtained result illustrates the applicability and usage of the MCDM approach in model selection.

Keywords: Reinforcement Learning (RL), Models, Fuzzy PROMETHEE, Decision Making, Stock Trading, Stock Market, MCDM

1. Introduction

The stock market plays a critical role in the overall financial market [1]. For a long time, researchers have been trying to figure out how to get useful trading signals during the transaction process in order to maximize the gains. The financial markets' price predictions are a hot topic in today's research as researchers look for reliable models that are simple to deploy, and are consistent in predicting accurate signals for trading the stock market to optimize profits and returns. Stock price prediction is one of the most challenging tasks in the field of financial market forecasting [2,3]. Technical analysis and fundamental analysis are the two main methods used to evaluate and forecast stock prices [1,2,4]. But this has been greatly challenged. The technical analysis only looks at past market data to forecast the future. Contrarily, the fundamental analysis considers additional data such as the condition of the economy, headlines, financial statements, meeting notes from discussions between Chief executives, etc. The efficient market hypothesis is a foundation for technical analysis [1,2,4,5]. According to the efficient market hypothesis, stock prices will quickly react to market fluctuations. In reality, the price can change in a matter of milliseconds, resulting in extremely high stock liquidity [4]. Technical analysis has received a lot of attention recently for the straightforward reason that we can gather enough information by simply looking at the historical stock market, which is open to the public and well-organized, as opposed to fundamental analysis, where we must examine unstructured datasets [1,2,4,6]. Both technical and fundamental analysis performed by humans has been greatly challenged by the inability to consistently optimize profits returns and the prediction of future outcomes [1,2,4,6]. The major goal of stock trading is to maximize returns while trying to avoid high risks[6,7].

Given the rapid growth of the deep learning community, deep learning techniques have recently been the most popularly chosen techniques because it is believed that deep reinforcement learning algorithms can outperform human players and other traditional statistical learning algorithms [1,2,8]. The non-stationarity and non-linearity of the stock markets are other factors that traditional statistical learning algorithms cannot handle [1,2]. The need to create something novel has arisen as modern artificial intelligence techniques have gotten closer to how people think and act. Human cognition and learning are stimulated by deep reinforcement Learning, which combines the perception of deep learning with the capacity for decision-making of reinforcement learning. This technique can output actions directly through the simulation of a deep neural network, which can be directly controlled according to the input image without the need for external constant monitoring, and it can input vision and other high-dimensional and multidimensional resource information[1,2]. By extracting the input data from the higher dimension, a deep neural network can automatically locate the corresponding representation of the lower dimension. Integrating respondent bias into the hierarchical neural network architecture is at the heart of deep learning [1,2]. Deep Learning and reinforcement learning, therefore, have strong feature extraction and perception capabilities [1,2,9]. Its weakness is that it is incapable of making decisions [5,8,10]. Reinforcement learning can be used directly for decision-making, i.e. to decide how to buy, hold or sell any stock. But it has difficulties fully expressing perception [1,2,4–8,10,11]. Stock price prediction and stock trading are the two main uses of deep learning and reinforcement learning in the stock markets [1,2]. Price regression and stock trend prediction are the two subsets of applications for stock price prediction. In the first application, numerical prices are precisely predicted, typically using a stock's closing price or day-wise price. In the second method, the turning point of a stock price, or when it changes direction from up to down or vice versa, is typically predicted [12]. Due to the stock market's non-stationary and non-linear nature, traditional methods of stock market forecasting based on fundamental and technical analysis are typically difficult. Deep learning, both supervised and unsupervised techniques, have been utilized to combine fundamental and technical analysis for stock price prediction and stock trading [7,8,12]. Numerous studies using reinforcement learning have been reported in literature [1,2,7,8,10–16].

It is already a known fact that reinforcement learning models can be deployed to predict the stock market, but the question is, which model is more effective and reliable? Which model is simple to deploy? Which model is stable and consistent in profit optimization? These questions have been left unanswered even though many existing works of literature have deployed different reinforcement models with their respective potentials. This has motivated us to evaluate common reinforcement learning models that have been deployed in different literature for stock trading based on well-defined reinforcement algorithms, and then, we compared them based on their performance using a hybrid multicriteria decision-making method called Fuzzy PROMETHEE, by establishing our comparison based on important criteria and factors that determine the applicability of reinforcement learning models for the prediction of the stock exchange market. This integrated approach is a unique approach that can provide a scheme for the construction of a sophisticated cognitive decision-making system for reinforcement learning models. Additionally, no existing research has integrated this approach to evaluate, compare, and rank reinforcement learning models used in trading stock.

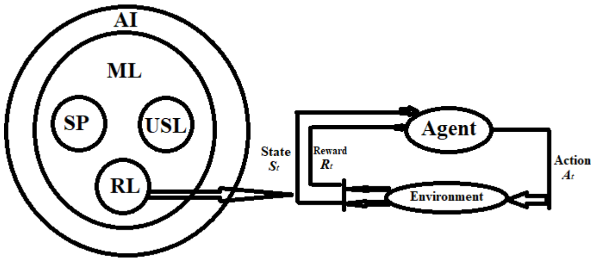

Fig. 1: The relationship between Artificial Intelligence (AI), Machine learning (ML), Supervised Learning (SL), Unsupervised Learning, and Reinforcement Learning (RL)
The diagram of the relationship between Artificial Intelligence (AI), Machine learning (ML), Supervised Learning (SL), Unsupervised Learning, and Reinforcement Learning (RL) is shown in Figure 1.

1. Reinforcement Learning in Stock Trading

Reinforcement learning as visualized in Figure 1, is a subset of machine learning [17,18]. Reinforcement learning learns how to take what kind of actions are suitable for a particular situation in order to maximize rewards to reach a specific goal [1,2]. It is different from supervised and unsupervised learning in way that supervised learning learns to predict from corresponding labels or output values associated with it, while unsupervised learning learns the underlying patterns or distributions that govern a given set of data. But in reinforcement learning, the agent (i.e. a piece of software you are training) learns through discovering actions that yield the most rewards through its experience. This is only done when agents repeatedly interact with the environment (i.e. the surrounding area where the agent interacts) [19]. Reinforcement learning has been used in playing games, wind energy optimization, industrial robotics, video game designs, fraud detection, autonomous driving, and in stock trading. Reinforcement learning has been very effective in stock trading [15,19,20]. As seen in Figure 1, consider the agent to be a stock trader, and the environment to be the stock market. The agent takes an action and is rewarded at time step t. Then, the environment changes to a new state. The agent must learn how to respond to its environment so that it can maximize its overall reward [1]. The models analyzed in this study are depicted in Figure 2.

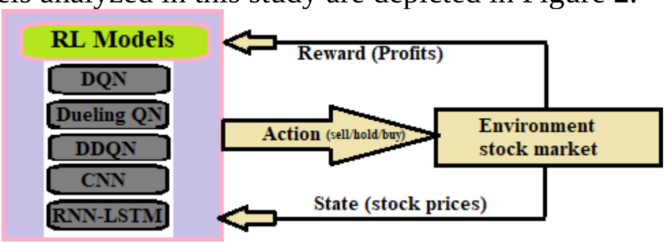

Fig. 2: Overview of RL models in stock trading

1.1. Deep Q Network (DQN)

One of the most well-known and effective reinforcement learning algorithms is Deep Q Network (DQN) [1,2,15]. It is a multi-layered neural network that generates a vector of action values for a given input state [2]. DQN is a particular type of network that employs a neural network to forecast Q value and continuously modifies the neural network to discover the maximum Q value. In the DQN, there are two neural networks: The Target Network, which is used to obtain the target value and has relatively fixed parameters, and the Current Q-Network, which assesses the current Q value. The actions (a), rewards (r), and outcomes of the next state (s, a, r, and s′) are recorded in replay memory, from which the training data is randomly taken [1,2]. Networks regularly update their parameters in response to environmental changes, and replay memory does the same. The Q value in DQN stands for the most recent learned experience. Learning the q-value function is essential to the DQN model in order to converge and successfully predict the Q value of each action in a range of states [2]. A study by [21] developed a DQN that is able to combine reinforcement learning with a class of artificial neural networks to evaluate the performance efficiency of DQN over others in games. The Atari 2600 platform which offers a diverse array of tasks, and is difficult for human players was used to evaluate the DQN agent. The DQN was compared with other efficiently performing reinforcement learning models along with a human game tester that is highly proficient in playing under controlled conditions. The study showed that the DQN method outperforms the best existing reinforcement learning methods. Another study by [22] developed a DQN for automated stock trading to make automatic decisions and achieve long-term stable profits. When DQN was compared with benchmarks of buy and hold and random action-selected DQN trade, the results showed that DQN outperforms the benchmarks. Other two classic models in deep reinforcement learning modified by the DQN models are; Double Deep Q-Network (DDQN), and Dueling Double Deep Q-Network (Dueling DDQN).

1.2. Double Deep Q-Network (DDQN)

DDQN is the combination of an old neural network and a new neural network, where the new neural network has an updated internal parameter with a time difference [2,23]. The DQN optimal Q value has been known to do the selection and evaluation of actions, this has often led to choosing an overestimated value which usually leads to an overestimation of the Q value. This overestimation of the Q value leads to an accumulated error with the increase in the number of iterations. Van Hasselt et al. [21] proposed the DDQN model to solve this overestimation problem. In the DDQN model, one of the Q networks chooses the action and the other evaluates the action. The new neural network helps to optimize the influence of error and solve the deviation problems that exist in DQN by modifying the generating of the target Q value [23]. A study by [22] showed that DDQN outperforms human beings in many fields such as playing Atari games and also in making trading decisions. When DDQN was compared to other proficient models, the results showed that the DDQN outperforms all models and even the DQN model. The DDQN model is able to discover and exploit profitable patterns more than other models. A study by [23] showed that the DDQN outperforms the DQN both in accuracy and policy quality. In trading stock, [24] proved the effectiveness of DDQN in predicting accurate trade signals and executing trade positions. Another study by [25] also proved that DDQN is able to solve the overestimation problems of DQN and therefore, it is a more robust model in reinforcement learning. Finally, Kim et al. [26] performed a comparative study and compared the performance efficacy of the DDQN model with the DQN model in stock trading. Results from the study showed that DDQN outperforms DQN and guaranteed increased and stable trading returns.

1.3. Dueling Q-Network

This dueling network is a single Q network with two streams that substitutes the typical one-stream Q network in existing techniques like Deep Q-Networks. Without any additional supervision, the dueling network automatically generates independent estimations of the state value function and advantage function. Depending on the impact of various actions, the value functions of the state action pairs in many DRL functions vary. The size of the value function, however, may differ depending on the state in some cases. In light of this, Wang et al. [27] suggested adding Dueling DQN to the DQN network pattern. Dueling DQN combines DQN and Dueling Network [27]. The performance capacity of the dueling Q-network was presented in a study by [28] based on 10 Indian stock datasets. The dataset contained the trade histories, and trade volumes of index NIFTY 50. The result from the study shows that the dueling Q-network model outperforms the DDQN and DQN models. Another study by [29] also proved that the dueling Q-network model is an efficient model in reinforcement learning which can be deployed in intraday trading of the stock market.

2.4. Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) are one kind of deep learning network that perform best at image processing tasks [30]. CNN models can be used to generate feature map visualizations to determine where the neural network is placing its attention on the candlestick images. CNN can switch its attention from all the candles in a candlestick image to the more recent ones in the image based on an event in the trading market. Computer vision and image classification tasks have both made extensive use of this network. In order to convert a pixel to a signal and train AI to play the game, CNN is also used. The result is a classification of different signal types. A convolutional layer and a subsampling layer are the two different types of layers that make up CNN [30,31]. These various layers will connect one after the other. Convolution will be performed in the convolutional layer, and the results will be passed on to the following layer. The representation size and parameter will be decreased until the data are a one-dimensional vector in the subsampling layer. CNN has proven to be very efficient in stock trading. A study by [30] trained a CNN model to make stock predictions. Preprocessed stock data were input into the model for an improved result of the model. The result from the study indicated that the CNN model is a robust model that can be deployed in making predictions in stock trading. Another study by [31] proposed an algorithmic CNN-TA trading model using a 2-D CNN that has a high image processing capacity. When compared with other common trading systems, the result indicated better performance for the CNN-TA model in buying, holding, and selling stock instruments. Finally, a study by [32] and [33] also proved the performance efficiency of the CNN models over LSTM and other common trading strategies [34].

2.5. Recurrent Neural Network (RNN)- Long Short-Term Memory Model (LSTM)

A Recurrent Neural Network (RNN) is a sort of NN that uses previous layers/information to extract current information and predict future trends [35,36]. To predict future trends, the RNN recalls the earlier extracted and stored information, in this case, the hidden layer serves as a repository for historical data from the sequential data. Due to the difficulties in storing long-term memory for RNNs, long short-term memory is used (LSTM) [37]. The memory line-based LSTM performed exceptionally well in forecasting scenarios including protracted data. An LSTM contains gates along the memory line that can be used to memorize previous information. The LSTM is a unique type of RNN because it can memorize data sequences [36]. A set of cells responsible for storing passed data streams must be present in every LSTM node. LSTM is one sort of RNN that can capture data from earlier stages and use it to make predictions for the future. The RNN-LSTM model has been very assistive in predicting the stock market. A research study by [38] and [39] deployed the LSTM model for predicting stock prices. The result showed that the LSTM model performed very well in generating profits. Another study by [40] presented the RNN-LSTM model to deal with anticipated stock market files. Results proved to be very efficient with the LSTM model. The performance accuracy was equated to about 97%. Finally, a research study by [41] optimized the LSTM model to prove its feasibility and performance in generating trade signals. When tested with six U.S market stocks, an average accuracy of 59.5% was obtained. The model was able to generate a total profit of $4143,233.33 with a $6,000,000 initial investment capital.

2. Methodology

The evaluation, assessment, and comparison of models have historically been based on performance evaluation metrics like mean absolute percentage error (MAPE), accuracy, F1 score, log loss, precision, recall, specificity, and so forth. None have increased the number of significant metrics or performance evaluation metrics to evaluate models that are more reliable, flexible, and less compromising. Consequently, there are several research questions, such as what happens if a decision-maker requires significant key aspects that are not covered by performance evaluation metrics.

This research study takes a novel approach to evaluate reinforcement models used in stock trading by using the Multicriteria decision-making method (MCDM) called fuzzy PROMETHEE based on certain selected criteria. This methodology has never been deployed in evaluating reinforcement learning models in stock trading. Therefore, this method is unique in its kind to this study.

Performance metrics must be used to evaluate the model's predictive ability after reinforcement learning models have been developed. Accuracy and precision in performance are the focus of these metrics. However, none mentions other crucial aspects including the model's applicability, functioning, and the effects of different factors on the model. The question of whether an "accurate model" can manage redundant and irrelevant market variables and if a precise model can be applied to a large dataset can be reduced to this. These are crucial factors for decision-makers to consider when selecting a model. Examples of these include the number of training samples needed, the effect of feature scaling, the effect of hyperparameter adjustment, and sensitivity to trivial features. MCDM approaches are crucial in this regard. One of the most important ways to choose the optimal course of action from a variety of options is to use MCDM approaches. It is a powerful tool with tremendous potential in the field of operational research that deals with how to compare a group of options using a variety of criteria [42–44]. We suggest combining and assessing the predictive, adaptability, and usability criteria of reinforcement learning models using MCDM. As a result, decision-makers will have access to resources that will help them make informed decisions when choosing the ideal model for stock trading [45].

3.1 Application of Fuzzy PROMETHEE

PROMETHEE is an MCDM technique that is user-friendly. It can be perfectly applied to real-life problem structures and is known for its efficiency in providing more preferences to decision-makers and fuzzy logic supports the decision-makers considering uncertainty based on available criteria in the PROMETHEE model [44]. PROMETHEE 1 is a partial ranking structure and PROMETHEE II is the complete ranking structure and (both), is a technique that provides simplicity for ranking the alternatives.

In this study, several criteria were proposed and weights of importance were assigned to each criterion based on expert opinions to evaluate the alternatives. The criteria include; accuracy, precision, consistency in making profits, simplicity in implementation, profit optimization rate, volatility rate, reliability, and speed. To implement fuzzy PROMETHEE, each criterion is simplified using a linguistic scale of relevance as seen in Table 1. RL models were evaluated using the selected criteria and their importance weights as shown in Table 2 using the fuzzy linguistic scale. In addition, the Yager index was applied to de-fuzzified the fuzzy values using Equation 1.

$$(3N - a + b)/3 \qquad (1)$$

where N is the center of the set, a is the distance between the center and left bound and b is the distance between the center and the right bound.

The Yager index is a recommended technique for defuzzification since it considers all possible

points of the sets for this process [45]. Finally, the PROMETHEE approach was deployed using the Gaussian preference functions for each criterion.
There are 5 main steps of the PROMETHEE method to be applied for the MCDM analysis

Step 1: The preference function Pj(d) of each criteria j should be defined.
Step 2: Importance weights of each criteria wt=(w1, w2, …, wk) should be defined.
Step 3: For each of the alternative pairs a_t, $a_{t'}$ ∈ A, the outranking relation (π) should be determined by the:

$$\pi(a_t, a_{t'}) = \sum_{k=1}^{K} w_k . [p_k(f_k(a_t) - f_k(a_{t'}))] , \; AXA \rightarrow [0,1] \quad (2)$$

where π (a, b) denotes the preference indices, which shows the preference intensity for an alternative a_t in comparison to an alternative $a_{t'}$ while counting all criteria.

Step 4: The positive and negative outranking flows should be determined as follows:

A positive outranking flow of the alternative a_t:

$$\Phi^+(a_t) = \frac{1}{n-1} \sum_{\substack{t'=1 \\ t' \neq t}}^{n} \pi(a_t, a_{t'}) \quad (3)$$

A negative outranking flow of the alternative a_t:

$$\Phi^-(a_t) = \frac{1}{n-1} \sum_{\substack{t'=1 \\ t' \neq t}}^{n} \pi(a_{t'}, a_t) \quad (4)$$

n denotes the number of the alternatives. The $\Phi^{+}(a_t)$ defines the strength of alternative $a_t \in$ A, while the negative outranking flow $\Phi^{-}(a_t)$ defines the weakness of alternative $a_t \in$ A.

PROMETHEE I determine the partial pre-order of the alternatives while PROMETHEE II determines the net ranking to alternatives. The partial pre-order of the options can be determined based on the following statements:

Via PROMETHEE I, alternative a_t is selected to alternative $a_{t'}$ ($a_t P a_{t'}$) if it satisfies either of the statements given below.

$$\begin{cases} \Phi^{+}(a_t) \geq \Phi^{+}(a_{t'}) \; and \; \Phi^{-}(a_t) < \Phi^{-}(a_{t'}) \\ \Phi^{+}(a_t) > \Phi^{+}(a_{t'}) \; and \; \Phi^{-}(a_t) = \Phi^{-}(a_{t'}) \end{cases} \tag{5}$$

a_t is indifferent to alternative $a_{t'}$ ($a_t I a_{t'}$) if:

$$\Phi^{+}(a_t) = \Phi^{+}(a_{t'}) \; and \; \Phi^{-}(a_t) = \Phi^{-}(a_{t'}) \tag{6}$$

And at is incomparable to $a_{t'}$ ($a_t R a_{t'}$) if:

$$\begin{cases} \Phi^{+}(a_t) > \Phi^{+}(a_{t'}) \; and \; \Phi^{-}(a_t) > \Phi^{-}(a_{t'}) \\ \Phi^{+}(a_t) < \Phi^{+}(a_{t'}) \; and \; \Phi^{-}(a_t) < \Phi^{-}(a_{t'}) \end{cases} \tag{7}$$

Step 5: The net outranking flow can be calculated for each alternative by using the Eq. (8).

$$\Phi^{net}(a_t) = \Phi^{+}(a_t) - \Phi^{-}(a_t) \tag{8}$$

Via PROMETHEE II, the complete order with net flow can be determined as:

a_t is preferred to $a_{t'}$ ($a_t P a_{t'}$) if $\Phi^{net}(a_t) > \Phi^{net}(a_{t'})$ (9)

a_t is indifferent to $a_{t'}$ ($a_t I a_{t'}$) if $\Phi^{net}(a_t) = \Phi^{net}(a_{t'})$ (10)

The higher $\Phi^{net}(a_t)$ value provides the better alternative.

The criteria that were used to evaluate alternatives during the decision-making process were carefully chosen. It is necessary to give weights in order to determine the relative significance levels of each criterion because not all criteria are equally relevant. The most crucial criteria are given more weight, while the least crucial criteria are given less weight. The fuzzy PROMETHEE approach relies on the applied criteria, weighted criteria, and defined preferences to rank particular alternatives. Different decision-makers may have different preferred alternatives and criteria and the outcomes can be updated accordingly. Diverse decision-makers may come up with different ideas based on predetermined preferences to compare, analyze, and rank outcomes when the necessity to choose criteria occurs. Expert opinion is crucial and required to obtain the most ideal solution to selection problems containing multiple parameters.

Knowing whether or not a reinforcement learning model predicts trade signals correctly is critical because it will significantly affect profit optimization. If a model is not consistent in producing correct signals and accurate rewards, a decision-maker will not want to start the deployment of the model [46] [47]. A decision-maker will also be interested in knowing the number of incorrect predictions generated by the model. When analyzing RL models used in stock trading, some of the most often utilized evaluation metrics include accuracy, reliability, precision, and consistency in optimizing profits. They serve as the primary performance indicators for the model, highlighting successfully and erroneously classified values. As a result, they were assigned a very high weight as shown in Table 1. The rate at which profits are optimized is also important because no decision-maker will like to deploy a model that generates negative returns. Thus, the profit optimization rate was assigned a high weight. Some instruments in stock trading are known to have high volatility/liquidity rates, common examples are the National Association of Securities Dealers Automated Quotations (Nasdaq

100) and the Standard & Poor's 500 Index (S & P 500). A slow model will not be able to accurately generate profitable signals. Therefore, volatility rate/speed significantly impacts model performance and was also assigned a medium weight.

Table 1. Linguistic Fuzzy Scale and assigned weights of importance to the criteria.

Linguistic scale for ranking	Triangular Fuzzy Scale	Importance ratings of criteria
Very High (VH)	(0.75, 1, 1)	Accuracy, precision, consistency in making profits, reliability
High (H)	(0.50, 0.75, 1)	Profit optimization rate
Medium (M)	(0.25, 0.50, 0.75)	Volatility rate/Speed
Low (L)	(0, 0.25, 0.50)	
Very Low (VL)	(0, 0, 0.25)	

Table 2: Data set for evaluating RL models

Aim	Max	Max	Max	Max	Max	Max
Alternatives/ Criteria	**Accuracy**	**Precision**	**Consistency in making profits**	**Reliability**	**Profit optimization rate**	**Volatility rate/Speed**
DQN [1,2,15]	VH	H	H	YES	H	M
DDQN [2,23]	VH	H	VH	YES	VH	VH
Dueling QN [27]	H	H	M	NO	H	H
CNN [30,31]	M	M	H	YES	M	H
RNN-LSTM [34,35]	M	M	M	YES	H	H

3. Results and Discussions

DDQN outperformed other models with the highest accuracy, precision, reliability, consistency in profit optimization, and speed whereas naive CNN and RNN-LSTM have the lowest accuracy, precision, profit optimization, and speed. The results obtained were satisfactory. This makes the RL models entirely appropriate and satisfactory to implement in predicting the stock market. When compared with previous studies employing the models in stock trading, our approach for ranking RL models is reliable in decision-making.

With a net flow of 0.0823, DDQN was determined as the most favorable and preferred RL model in stock trading using the fuzzy PROMETHEE method of decision-making. DQN, Dueling QN, and CNN came second, third, and fourth, with net flows of 0.0364, −0.0142, and −0.0465, respectively. RNN-LSTM with a net flow of −0.0581 was the least preferred alternative, as shown in Table 3. However, the results may differ if a different weight is assigned to the criteria.

Table 3: PROMETHEE Flow Table

Rank	Alternatives	Outranking NetFlow	Positive NetFlow	Negative NetFlow
1	DDQN	0.0823	0.0823	0.000
2	DQN	0.0364	0.0451	0.0087
3	Dueling QN	-0.0142	0.0189	0.0331

4	CNN	-0.0465	0.0078	0.0542
5	RNN-LSTM	-0.0581	0.0073	0.0654

Fig. 3 displays the evaluation results of the models, highlighting their advantages and disadvantages as well as the final order of available options. Each model is represented in this graph from most to least preferred. The parameters above the 0 threshold denote the advantages of the alternative, while the parameters below the 0 threshold denote the disadvantages of those alternatives. The net flow values are shown in the diagram, where options are arranged from left to right according to rank. A vertical bar made up of criteria shows the alternatives. This bar illustrates the contribution of each criterion to the final net flow value of an alternative. The height of the vertical bar, multiplied by the appropriate weight of the given criterion, displays the difference between the positive and negative preference flow. The highest positive values are displayed by the indications at the top of the vertical bar, while the highest negative values are displayed by the indicators at the bottom of the vertical bar. As a result, the PROMETHEE diagram offers a thorough picture of all options and requirements, together with an assessment of their relative weight [48].

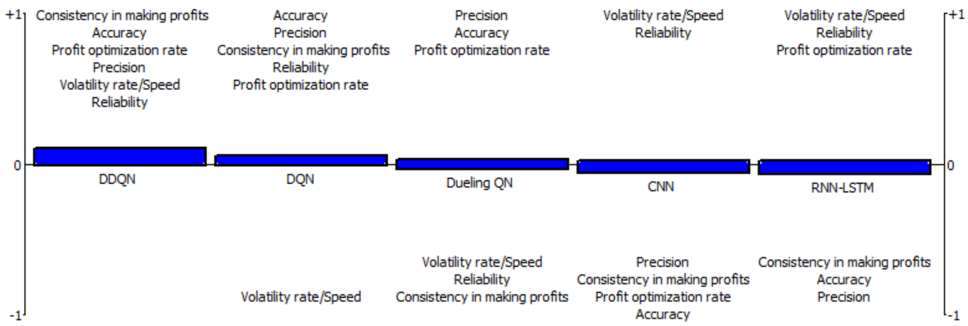

Fig. 3: PROMETHEE Evaluation ranking of RL algorithms.

Conclusion

This study suggests a novel method for selecting the best RL model for signal generation and prediction in stock trading. By including more variables than only the often-used key metrics, this innovative technique advances the evaluation of RL models and thereby creates a new path for model evaluation. Important factors including accuracy, precision, consistency in producing profits, ease of implementation, profit optimization rate, volatility rate/speed, reliability, and speed were considered in this study. These criteria are important, as demonstrated by the study's findings. With this study, existing literature relating to RL models for stock trading has been verified, and it is aimed to inform stock traders that are uncertain about the best RL models for predicting the stock market.

The findings of this study show that the deployed method is useful and effective for evaluating RL model performance. The result might change if the weights given to the various criteria are changed. The obtained result illustrates the applicability and usage of the MCDM approach in model selection.

Acknowledgment: I thank **Assoc. Prof. Dr. Berna Uzun** and **Prof. Dr. Dilber Uzun Ozsahin** for reviewing this work and making sure everything is correct.
Conflicts of Interest: There is no conflict of interest between the authors.
Funding: There is no funding.

References

[1] “Reinforcement Learning in Stock Trading”, Accessed: Oct. 01, 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02306522

[2] Y. Li, P. Ni, and V. Chang, “Application of Deep Reinforcement Learning in Stock Trading Strategies and Stock Forecasting”.

[3] E. Kiral and B. Uzun, “Forecasting Closing Returns Of Borsa Istanbul Index With Markov Chain Process Of The Fuzzy States,” Journal of Economics, Finance and Accounting-JEFA, vol. 4, no. 1, pp. 15–24, 2017, doi: 10.17261/Pressacademia.2017.362.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” Jul. 2017, Accessed: Oct. 01, 2022. [Online]. Available: https://towardsdatascience.com/deep-reinforcement-learning-for-automated-stock-trading-f1dad0126a02

[5] B. M. Henrique, V. A. Sobreiro, and H. Kimura, “Literature review,” Expert Syst Appl, vol. 124, pp. 226–251, Jun. 2019, doi: 10.1016/J.ESWA.2019.01.012.

[6] “Stock Market Prediction, The Planetary Barometer and How to Use It (Reprint of 1948 Edition) by Bradley, Donald A.: Very Good Soft cover (1948) Reprint | Alanpuri Trading.” https://www.abebooks.com/Stock-Market-Prediction-Planetary-Barometer-Use/9811061811/bd (accessed Oct. 01, 2022).

[7] Y. Zuo and E. Kita, “Stock price forecast using Bayesian network,” Expert Syst Appl, vol. 39, no. 8, pp. 6729–6737, Jun. 2012, doi: 10.1016/J.ESWA.2011.12.035.

[8] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “NSE Stock Market Prediction Using Deep-Learning Models,” Procedia Comput Sci, vol. 132, pp. 1351–1362, Jan. 2018, doi: 10.1016/J.PROCS.2018.05.050.

[9] M. Ballings, D. van den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” Expert Syst Appl, vol. 42, no. 20, pp. 7046–7056, Jun. 2015, doi: 10.1016/J.ESWA.2015.05.013.

[10] X. dan Zhang, A. Li, and R. Pan, “Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine,” Appl Soft Comput, vol. 49, pp. 385–398, Dec. 2016, doi: 10.1016/J.ASOC.2016.08.026.

[11] E. J. Elton, M. J. Gruber, S. J. Brown, and W. N. Goetzmann, “Modern portfolio theory and investment analysis,” p. 738, Accessed: Oct. 01, 2022. [Online]. Available: https://www.wiley.com/en-us/Modern+Portfolio+Theory+and+Investment+Analysis%2C+9th+Edition-p-9781118469941

[12] B. ; Krollner, B. ; Vanstone, G. Finnie, B. Krollner, and B. Vanstone, “Financial time series forecasting with machine learning techniques A survey Financial time series forecasting with machine learning techniques: A survey Financial Time Series Forecasting with Machine Learning Techniques: A Survey,” pp. 25–30, 2010, Accessed: Oct. 01, 2022. [Online]. Available: http://epublications.bond.edu.au/infotech_pubs/110

[13] F. Abtahi, Z. Zhu, and A. M. Burry, “A Deep Reinforcement Learning Approach to Character Segmentation of License Plate Images,” 2015.

[14] “Reinforcement Learning For Trading Applications.” https://alphaarchitect.com/2020/02/reinforcement-learning-for-trading/ (accessed Oct. 01, 2022).

[15] A. Brim and N. S. Flann, “Deep reinforcement learning stock market trading, utilizing a CNN with candlestick images,” PLoS One, vol. 17, no. 2, p. e0263181, Feb. 2022, doi: 10.1371/JOURNAL.PONE.0263181.

[16] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, “An Introduction to Deep Reinforcement Learning,” Foundations and Trends® in Machine Learning, vol. 11, no. 3–4, pp. 219–354, Dec. 2018, doi: 10.1561/2200000071.

[17] A. Leite, M. Candadai, and E. J. Izquierdo, “Reinforcement learning beyond the Bellman equation: Exploring critic objectives using evolution,” pp. 441–449, Jul. 2020, doi: 10.1162/ISAL_A_00338.

[18] P. Carrera Flórez De Quiñones, V. Laparra, P.-M. Jordi, and M. Marí, “Reinforcement

Learning in Stock Market”.
[19] J. W. Scholars and Y. Liu, “ScholarlyCommons ScholarlyCommons Reinforcement Learning Applications in Real Time Trading Reinforcement Learning Applications in Real Time Trading,” 2019, Accessed: Oct. 03, 2022. [Online]. Available: https://repository.upenn.edu/joseph_wharton_scholars/
[20] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature 2015 518:7540, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.
[21] L. Chen and Q. Gao, “Application of Deep Reinforcement Learning on Automated Stock Trading,” undefined, vol. 2019-October, pp. 29–33, Oct. 2019, doi: 10.1109/ICSESS47205.2019.9040728.
[22] H. van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, pp. 2094–2100, Mar. 2016, doi: 10.1609/AAAI.V30I1.10295.
[23] L. Chen and Q. Gao, “Application of Deep Reinforcement Learning on Automated Stock Trading,” undefined, vol. 2019-October, pp. 29–33, Oct. 2019, doi: 10.1109/ICSESS47205.2019.9040728.
[24] “DEEP REINFORCEMENT LEARNING IN ALGORITHMIC TRADING (Part- II) | by Astarag Mohapatra | Analytics Vidhya | Medium.” https://medium.com/analytics-vidhya/deep-reinforcement-learning-in-algorithmic-trading-part-ii-b78db754961c (accessed Oct. 04, 2022).
[25] M. T. Kim and K. B. Wook, “Trading Bot Implementation and Performance Comparison Using DQN and DDQN,” 전기학회논문지, vol. 70, no. 1, pp. 158–167, 2021.
[26] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Frcitas, “Dueling Network Architectures for Deep Reinforcement Learning,” 33rd International Conference on Machine Learning, ICML 2016, vol. 4, pp. 2939–2947, Nov. 2015, doi: 10.48550/arxiv.1511.06581.
[27] S. Bajpai, “APPLICATION OF DEEP REINFORCEMENT LEARNING FOR INDIAN STOCK TRADING AUTOMATION A PREPRINT,” 2021.
[28] L. Zhang et al., “You may also like Multi-input Convolutional Neural Network Fault Diagnosis Algorithm Based on the Hydraulic Pump Research on Image Classification Algorithm Based on Convolutional Neural Network Lihua Luo-Symptomatically Brain Tumor Detection Using Convolutional Neural Networks Stock Prediction Using Convolutional Neural Network”, doi: 10.1088/1757-899X/435/1/012026.
[29] O. B. Sezer and A. M. Ozbayoglu, “Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach,” Appl Soft Comput, vol. 70, pp. 525–538, Sep. 2018, doi: 10.1016/J.ASOC.2018.04.024.
[30] S. K. Chandar, “Convolutional neural network for stock trading using technical indicators,” Automated Software Engineering, vol. 29, no. 1, pp. 1–14, May 2022, doi: 10.1007/S10515-021-00303-Z/FIGURES/8.
[31] A. Moghar and M. Hamiche, “Stock Market Prediction Using LSTM Recurrent Neural Network,” Procedia Comput Sci, vol. 170, pp. 1168–1173, 2020, doi: 10.1016/J.PROCS.2020.03.049.
[32] K. Narang, M. Gupta, R. Kumar and A. J. Obaid, "Channel Attention Based on ResNet-50 Model for Image Classification of DFUs Using CNN," 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 2024, pp. 1-6, doi: 10.1109/INCET61516.2024.10593169.
[33] Z. Zou and Z. Qu, “Using LSTM in Stock prediction and Quantitative Trading”.
[34] A. Pandey, “Prediction of Stock Price using RNN’s LSTM-Based Deep Learning Model,” Int J Res Appl Sci Eng Technol, vol. 9, no. 8, pp. 2469–2474, Aug. 2021, doi: 10.22214/IJRASET.2021.37791.
[35] D. Singh, M. Gupta, and R. Kumar, “BGR Images-Based Human Fall Detection Using ResNet-50 and LSTM,” Lecture Notes in Networks and Systems, vol. 608, pp. 175–186, 2023, doi: 10.1007/978-981-19-9225-4_14.
[36] N. Mottaghi and S. Farhangdoost, “Stock Price Forecasting in Presence of Covid-19

Pandemic and Evaluating Performances of Machine Learning Models for Time-Series Forecasting," May 2021, Accessed: Oct. 06, 2022. [Online]. Available: http://arxiv.org/abs/2105.02785
[37] K. POTHUGANTI, "Long Short-Term Memory (LSTM) Algorithm Based Prediction of Stock Market Exchange," SSRN Electronic Journal, Jan. 2021, doi: 10.2139/SSRN.3770184.
[38] Q. Gao, Z. He, and T. Supervisor, "STOCK MARKET FORECASTING USING RECURRENT NEURAL NETWORK," 2016.
[39] E. Precious Onakpojeruo, B. Uzun, and D. Uzun Ozsahin, "Hydrogel-Based Drug Delivery Nanoparticles with Conventional Treatment Approaches for Cancer Tumors; A comparative study using MCDM Technique," 2022, doi: 10.21203/rs.3.rs-2116197/v1.
[40] E. Precious Onakpojeruo, B. Uzun, and D. Uzun Ozsahin, "Evaluation of the Treatment Alternatives for Spinal Cord Tumors Using Analytical Evaluation Models," 2022, doi: 10.21203/rs.3.rs-2009799/v1.
[41] B. Duwa, E. Precious Onakpojeruo, B. Uzun, and D. Uzun Ozsahin, "Comparative Evaluation of 3D Filaments, Used in Additive Manufacturing of Biomedical Tools; Using Fuzzy Promethee," 2022, doi: 10.21203/rs.3.rs-2020207/v1.
[42] I. Ozsahin, T. Sharif, D. U. Ozsahin, and B. Uzun, "Evaluation of solid-state detectors in medical imaging with fuzzy PROMETHEE," Journal of Instrumentation, vol. 14, no. 01, p. C01019, Jan. 2019, doi: 10.1088/1748-0221/14/01/C01019.
[43] V. Ahuja, U. Singh, M. Gupta, and R. Kumar, "A Critical Analysis of Artificial Intelligence in Stock Market Prediction: A Literature Review," NEU Journal for Artificial Intelligence and Internet of Things, vol. 1, no. 2, Oct. 2023, Accessed: Dec. 26, 2024. [Online]. Available: https://dergi.neu.edu.tr/index.php/aiit/article/view/798
[44] Di. U. Ozsahin, K. Nyakuwanikwa, T. Wallace, and I. Ozsahin, "Evaluation and Simulation of Colon Cancer Treatment Techniques with Fuzzy PROMETHEE," 2019 Advances in Science and Engineering Technology International Conferences, ASET 2019, May 2019, doi: 10.1109/ICASET.2019.8714509.
[45] D. Kumar, M. Gupta and R. Kumar, "Tata Steel Stock Forecasting Using Deep Learning," 2023 2nd International Conference on Computational Modelling, Simulation and Optimization (ICCMSO), Bali, Indonesia, 2023, pp. 33-39, doi: 10.1109/ICCMSO59960.2023.00020.
[46] Gupta, M., Kumar, R., Chaudhary, R. K., & Kumari, J. (2021, December). IoT based voice controlled autonomous robotic vehicle through google assistant. In *2021 3rd international conference on advances in computing, communication control and networking (ICAC3N)* (pp. 713-717). IEEE.
[47] Gupta, M., Kumar, R., Walia, H., & Kaur, G. (2021, October). Airlines based twitter sentiment analysis using deep learning. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-6). IEEE.
[48] Kumar, R., Gupta, M., Ahmed, S., Alhumam, A., & Aggarwal, T. (2022). Intelligent Audio Signal Processing for Detecting Rainforest Species Using Deep Learning. *Intelligent Automation & Soft Computing*, *31*(2).