IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 October 2022, accepted 17 November 2022, date of publication 2 December 2022,
date of current version 9 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226629

RESEARCH ARTICLE

A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading

YASMEEN ANSARI[1], SADAF YASMIN[2], SHENEELA NAZ[3], HIRA ZAFFAR[4], ZEESHAN ALI[5], JIHOON MOON[6], AND SEUNGMIN RHO[7]

[1]Department of Finance, College of Administrative and Financial Sciences, Saudi Electronic University, Riyadh 13323, Saudi Arabia
[2]Department of Computer Science, COMSATS University Islamabad, Attock Campus, Attock 43600, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Islamabad 45550, Pakistan
[4]Department of Computer Science, Air University, Aerospace and Aviation Kamra Campus, Islamabad 44000, Pakistan
[5]Research and Development Setups, National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan
[6]Department of AI and Big Data, Soonchunhyang University, Asan 31538, South Korea
[7]Department of Industrial Security, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Seungmin Rho (smrho@cau.ac.kr)

This research was supported by the Chung-Ang University Research Grants in 2022 and also the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2018-0-01799) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

ABSTRACT Presently, the volatile and dynamic aspects of stock prices are significant research challenges for stock markets or any other financial sector to design accurate and profitable trading strategies in all market situations. To meet such challenges, the usage of computer-aided stock trading techniques has grown in prominence in recent decades owing to their ability to rapidly and accurately analyze stock market situations. In the recent past, deep reinforcement learning (DRL) methods and trading bots are commonly utilized for algorithmic trading. However, in the existing literature, the trading agents employ the historical and present trends of stock prices as an observing state to make trading decisions without taking into account the long-term market future pattern of stock prices. Therefore, in this study, we proposed a novel decision support system for automated stock trading based on deep reinforcement learning that observes both past and future trends of stock prices whether single and multi-step ahead as an observing state to make the optimal trading decisions of buying, selling, and holding the stocks. More specifically, at every time step, future trends are monitored concurrently using a forecasting network whose output is concatenated with past trends of stock prices. The concatenated vectors are subsequently supplied to the DRL agent as an observation state. In addition, the suggested forecasting network is built on a Gated Recurrent Unit (GRU). The GRU-based agent captures more informative and inherent aspects of time-series financial data. Furthermore, the suggested decision support system has been tested on several stock markets such as Tesla, IBM, Amazon, CSCO, and Chinese Stocks as well as equity markets i-e SSE Composite Index, NIFTY 50 Index, US Commodity Index Fund, and has achieved encouraging profit values while trading.

INDEX TERMS Decision support system, automated stock trading, deep reinforcement learning, deep-Q networks, forecasting network, GRU, long-term market future patterns.

I. INTRODUCTION

A stock market is a common place for the sellers and buyers of assets, stocks, or shares. To perform trading activities including selling and buying these stocks or shares, consumers can view stock exchanges [1]. In the context of stock market trading and analysis, the continuously increasing volume of financial data has far surpassed the ability of investors or decision-makers to manually interpret it. In comparison with statistical data, financial time series data is more

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

sophisticated. This is due to environmental factors including cyclical changes, seasonal fluctuations, and erratic movements. Moreover, various external variables, as well as many intricately interconnected economic, governmental, societal, and even psychological aspects, have a significant impact on them [2], [3]. This underlined the crucial need for automated ways of interpreting such volatile, enormous, and unpredictable data to obtain meaningful facts and statistics from them. For this financial time-series analysis, different data mining strategies have created their place to guide investors to make strategic, and knowledge-based decisions to maximize profits while minimizing financial risks [4]. Generally, the end goal of investors involved in the financial sector is to make more profits. To obtain such profits or gains, there exist many investment opportunities including trading i-e buying and selling, valuable metals e-g gold, shares, foreign currencies, and others. Trading is the most common kind of financial activity in the stock market. The best way of making large gains in trading stocks is by determining the best trading period with the least amount of risk. Given the unpredictable patterns of the stock market, it is very challenging to determine the decision between buying and selling stocks.

In the existing literature, different researchers proposed methods of technical analysis of financial data to guide investors in making the rules of trading for buy-sell-hold selections. Delving into more depth, different researchers have performed stock market analysis-related tasks in numerous ways. Generally, the technical analysis can be done by employing the past historical data of stocks to estimate the future trends of stocks which will ultimately assist the investor to take different decisions regarding stocks in the next stage. Currently, machine learning, data mining, artificial intelligence, and deep learning methods are extensively adopted methods in different applications areas of medical assistance [5], [6], security and surveillance [7], agriculture [8], [9], recommendation assistance [10], and many more [11]. But now some researchers have performed stock market prediction using statistical methods [12], traditional machine learning models [13], deep learning methods, and deep learning methods optimized by evolutionary algorithms [14]. All these approaches are also employed in financial sectors [15], [16]. These methods are utilized to compute the best trading signals through technical analysis. Gains and losses through trading stocks are determined solely by a study of the future trend of extremely volatile and erratic stock price elements. Effective categorization of rising and falling swings in stock price indicators may be beneficial not just to investors in developing an efficient trading plan, but also to policymakers in monitoring the financial markets.

However, the uncertainty present in the stock trends will make challenging to earn profit with human-made rules. The emotions of the traders are also a factor in losses during trading [17]. The fluctuation of stock values is so common that a human trader cannot always respond instantaneously [18], [19]. To overcome this, researchers proposed the concept of automated trading also known as Algorithmic Trading (AT) [20], [21]. This is referred to as a computer program that will perform trading according to the rules or trading logic designed by the programmer. The time needed to take trading decisions or transactions is also reduced when the comparison is done with human traders [22], [23]. In the existing literature, there are many trading strategies designed, for instance, Mean reversion [24]. More explicitly, some researchers designed the rules discovery method through which stock trading can be done [25]. Forecasting methods such as ensemble learning are designed to first forecast future price trends then based on those forecasted values the trading decision is made [26]. However, the same predefined strategies of trading are not often lucrative for all possible types of trends since they may be good for cases like an uptrend, downtrend, or sideways trend). Therefore, one of the major research problems in the area of stock trading is the adoption of optimum trading strategy from a variety of methods at a given point in time. It would be very helpful if the future trajectory of stock prices is available. However, this future estimation is also being affected by several environmental factors. Furthermore, trading techniques based on these predictions-based methods are static. In static techniques, when one trading plan is selected, it remains unchanged for the duration of the trading period [27]. Static tactics are high risk since stock market patterns are unclear and alter very rapidly. Hence, a flexible and dynamic trading strategy is essential that is adaptable to modifying its trading decisions in response to fluctuations in the stock market situation.

Hence, in this research study, instead of employing pre-determined strategies, we proposed a self-adjusting automated trading strategy as a decision support system for stock market investors by designing a deep reinforcement learning framework. For instance, in fundamental analysis financial reports and balance sheets are examined to generate trading signals and this can be accomplished by financial analysts. But with the advent of artificial intelligence and machine learning approaches, the difficulty of manually analyzing the reports for trading is eliminated with the help of such automated trading systems. Subsequently, behavioral analysis is the study of investor behavior and it was observed in existing studies that the behavior of investors has a strong impact on stocks as well as larger crash risks and financial decisions [28], [30]. One of the most prevalent variables includes overconfidence, the herding effect, overreactions, etc. However, the method proposed in this study has capable of reliability and quick decisions, as well as being unaffected by emotional, psychological, and cognitive variables. Although the technique presented in this paper is centered on technical analysis for stock trading, but it serves as a facilitating system for fundamental analysts and investors, and the combination of both analyses results in more profitable trading decisions. The main advantage of the DRL-based proposed model is that it will be applicable as a decision support system that updates itself depending upon the environment (market behavior). As in traditional algorithms, the trading rules are designed based on future forecasting of prices, and the rules remain

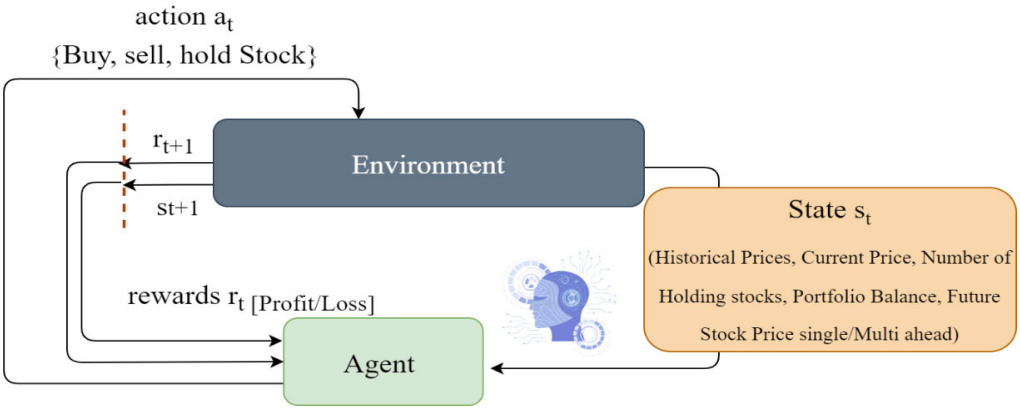


FIGURE 1. Formulation of automated stock market trading with reinforcement learning.

fixed and view the markets behaviors as static. Hence, the first contribution is to design adaptable trading strategies to overcome the challenges in traditional trading strategies. Moreover, there also exist some studies in which decision support systems are designed over the dynamic strategies i-e reinforcement learning but inputs to these agents (in form of states) are previous stock prices, and based on these values the agent makes the trading decisions [31], [32], [33]. One big concern emerges here: what if we made the agent aware of future events? (i-e. future prices of stocks are included as state information). Since the state's engineering of DRL-based agents has a very significant impact on their performance. The better the state representation the more accurate information is observed by the agent which will untimely assist in taking more optimal trading decisions. In the existing studies, the historical information of stock prices in the past is employed as state representation for the DRL agent's i-e $\boldsymbol{t-1}$ time steps. However, the stock market is very volatile in nature, therefore considering only past trends is not enough to take the optimal decisions regarding stock buying, selling, and holding. As a result, we design the forecasting network that estimates the future trends simultaneously i-e stock prices at $\boldsymbol{t+1}$ time steps. Subsequently, both historical and estimated future trends are employed to construct the states of the DRL agent. For future trends, this research takes into account both single-step (i-e instant following day future pricing because the data is on a daily basis) and multi-step future prices (i-e more than one-day e-g next two days). Through this, the DRL agent is aware of both past and future as shown in Figure 1 and by considering both types of information, it takes trading decisions by maximizing the total cumulative rewards in form of profit generated in the financial sector. As a result, providing both past and future trends as state information will help us with our second contribution. Thirdly, the forecasting network is based on Gated Recurrent Unit GRU. The GRU is generally more effective for financial time series data and is faster and less complex than other RNN-based neural networks i-e LSTM and Simple RNN. With this way and its characteristics, the proposed DRL agent becomes more effective in trading decisions. In addition, we have evaluated the proposed model on ten different stock datasets and the results indicate the effectiveness of the proposed model. The point-by-point contributions of this research study are listed below:

- A decision support system for automated stock market trading is designed based on deep reinforcement learning (DRL)
- Proposed DRL considers both past and future trends in stock prices as an observation state during stock trading decisions
- A GRU-based forecasting model is intended to assist the DRL agent when trading and making decisions
- To execute accurate algorithmic trading, the proposed RL agent is future aware in both single and multi-step contexts

The rest of the paper is categorized as: Section II describes the related work in stock trading systems, Section III shows the proposed model, while in Section IV the results of the proposed model are discussed followed by the conclusion and references.

II. RELATED WORK

This section discusses several existing strategies proposed by various researchers to optimize stock trading decisions using various approaches. These methods include different stock market trading strategies using TTRs (Technical Trading rules), machine learning, deep learning, and reinforcement learning methods.

Currently, different research studies exploited the use of TTRs methods for stock market trading. For instance, Metghalchi et al. [34] employs five TTRs methods such as moving average, relative strength index, momentum, etc. for trading decisions and evaluated them on the Turkish indexes. Likewise, Tudor et al. [35] exploits the use of TTRs in predicting the oil stock market by indicating the proper timing of market entrance and exit. Arif et al. [36] leverage portfolio return to use TTRs since portfolio trading serves as the most successful and widely used trading method in financial

markets worldwide. They have performed the experimentation on Pakistan stock changes. It is observed from these studies that TTRs based methods show good results in designing trading strategies, nevertheless, these are static methods and cannot be adapted according to environments.

Moreover, different machine learning methods, which are a subset of artificial intelligence (AI), are employed to assist investors and yield a greater return on equity than conventional analytical approaches [37].

Following on, Liu et al. proposed a new investment technique based on neural networks [38]. In this study, the results indicate that the suggested NN model supports the investors in reaching a decision by achieving 78% FScore in buying transactions and 60% in selling transactions. Gonzalez et al. proposed a trading system in which fundamental analysis is employed to strengthen the investment approaches [39]. A financial instrument named as relative strength index is used to produce the trading points which is the main premise of this study. The neural network is employed to compute the relative strength index. Similarly, Dash et al. employ the hybrid approach for stock trading in which technical analysis is accomplished with machine learning [4]. More explicitly, a collection of rules is defined to produce trading decisions. This problem of producing decisions is represented as the classification task in this study, in which three classes indicate buy, sell, and hold signals. They have performed the comparison with different algorithms such as the Naive Bayesian model, SVM, and KNN to show the effectiveness of their suggested model. Amanat et al. integrate several models of machine learning to carry out stock trading [40]. These models include Gaussian Naive Bayes, Decision Tree, and Logistic Regression. Their proposed model achieves 54.35% profit during trading between July 2011 and January 2019. Their suggested model is validated on the US stock market data. In addition, Zhang et al. use a reversion model to do stock market trading [41]. This model is reliant on XCS (extended Classifier Systems) which is a very appropriate model due to its inherent methodologies such as rules categorization mining, evolutionary learning, as well as RL that provide clear representative abilities.

With the advent of deep learning and reinforcement learning new concepts and ideas are introduced into the stock market. Regarding the stock trading challenge, there exist two lines of studies i-e deep learning and reinforcement learning as per distinct trading tactics. As seen in Figure 2, there are several types of reinforcement learning algorithms, such as Q-learning and deep-Q networks. The difference between algorithms is linked to the learning technique. The algorithms of deep learning, on either side, are employed to forecast the returns or stock trends in the market. For instance, in the work of Qiu et al. the daily trend of stock markets is predicted by integrating the neural network models with dimension reduction methods [42]. Subsequently, for the Japanese stock market, another model based on an artificial neural network is designed by Zhong et al. to predict the returns [43]. Deep learning techniques have also been utilized to forecast stock values. To maximize the advantage of the trading approach, the results of forecasting were included in quantitative trading methods. On another side, some researchers have exploited automated methods of trading to guide investors. For instance, Moody et al. devised a trading strategy based on reinforcement learning [44]. In this study, the input of the agent based on the neural network model is the raw financial data. Another study based on reinforcement learning is the work of Neves et al. in which a short-term speculating system is proposed in the foreign currency market [45]. Later on, models based on adaptive financial distress becomes a focus of the research studies of Sun et al. [46], [47] Their predictive models are effective for the administration of financial risks associated with companies. The representation of financial signals and trading is done with scattered coding in the research study of Yue et al. to develop an optimum trading system [48]. They built a trading plan by combining scattered or sparse coding with reinforcement learning, wherein sparse coding was utilized for extracting features and the model of reinforcement learning was the actor-only technique. Troiano et al. mimic the logic of a plan with an LSTM network (Long-Short-Term-Memory) employed to train a robot to perform trading decisions [49]. Yang et al. suggested an ensemble strategy of a reinforcement learning algorithm for stock market trading [50].

Therefore, the guidance to investor decisions is enhanced by employing AI-based approaches and also increasing profits by much more than 28% [37]. For instance, Boonpeng and Jeatrakul [37] designed a decision support system by suggesting One-Against-All (OAA) neural network to carry out investments in the stock market. Their suggested method shows that OAA techniques generate remarkable results in comparison with traditional methods with the greatest return rate of 57.67%. The algorithm is based on an actor-critic framework involving Deep Deterministic Policy Gradient (DDPG), Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). Xiong et al. also proposed an automated trading system by using the more practical approach of deep reinforcement learning [51]. The proposed framework is also based on the actor-critic framework namely, Deep Deterministic Policy Gradient (DDPG). In this study, trading equities are chosen from a pool of 30 stocks, and their daily prices serve as the training for the models and trade market situation. In respect of the Sharpe ratio and cumulative returns, their suggested deep reinforcement learning strategy outperforms the existing methods. Nan et al. proposed the deep reinforcement learning model by incorporating sentiments and knowledge to carry out the automated trading of stocks [52]. More explicitly, the proposed method employs the integrated data of stocks and sentiments regarding news headlines along with the utilization of knowledge graphs. Overall, neural network models have been shown to be productive and competitive in predicting stock prices and making investment choices.

Moreover, some fuzzy rules-based systems are also employed for the problem of stock market trading. For

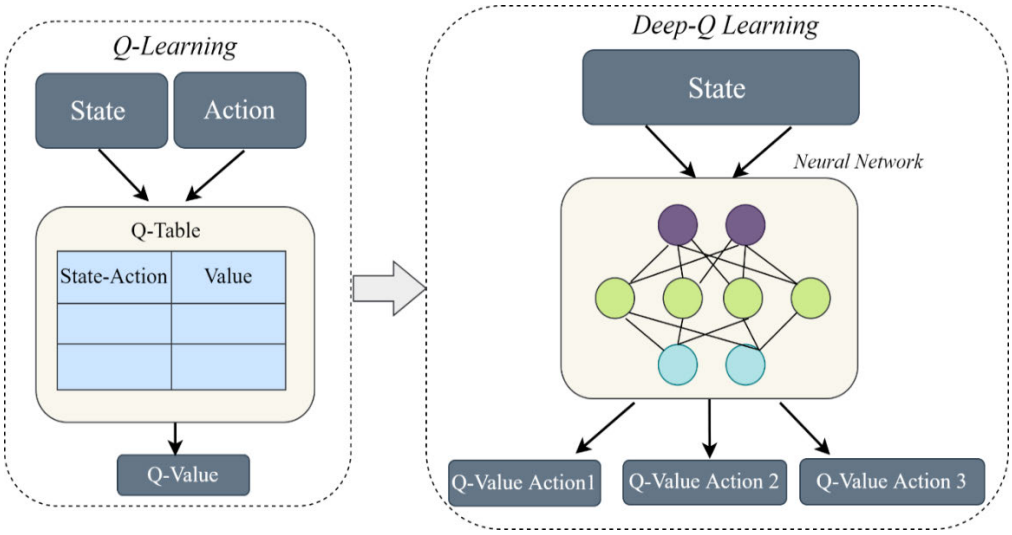


FIGURE 2. Pictorial representation of Q-learning and Deep-Q Networks.

instance, a high-order fuzzy model is designed to generate the decision rules as a financial guideline for investors in the study of Chen [53]. Furthermore, their model is reliant on entropy categorization and an adjustable expectation framework. Similarly, Sandy et al. suggest fuzzy logic controllers for the purpose of stock market trading [54]. In order to determine the robustness of trading signals including buy, sell, and hold candlestick characteristics and Bollinger Bands (BB) were utilized as a technical measure. The collection of rules is developed by fuzzy logic in which the indicators are produced reflecting the intensity of the execution decision. Subsequently, in the study of Aleksandar et al. in which clustering method is employed to perform stock trading [55]. The clustering is done by using a mathematical model designed over interpolative Boolean algebra. Following on, Kim et al. suggest a hybrid trading system in which trading rules are designed using rough sets and genetic algorithms [56]. Experiments were conducted on past data from the Korea Composite Stock Market Index 200 (KOSPI 200) futures trading to assess the proposed system. In all of the above-mentioned literature, starting from TTRs, machine learning, deep learning, reinforcement learning, fuzzy logic, evolutionary algorithm, and clustering methods are the different methodologies adopted by several researchers. However, the use of deep reinforcement learning is more accurate and effective since this model is adjustable according to the market situation.

As a result, in this work, we used deep reinforcement learning for stock market trading and modified the model to boost performance even more.

III. METHODOLOGY

In this section, we have explained the proposed model that acts as a decision support system for automated stock trading. The pictorial representation of the proposed work is depicted in Figure 3. In the first step, we collected the data from different stocks. Subsequently, we have designed a deep-Q Network-based reinforcement learning agent that observes the stock market situation which includes the historical trends of the stock prices. In addition, the future situation of stock prices is estimated through another deep learning model and is also observed by the agent to more accurately take the trading decisions. The RL agent maps the state to action-value pairs or takes trading decisions. It is built upon simple dense layers to fully observe the financial time series data.

A. REINFORCEMENT LEARNING

Reinforcement learning is a subfield of machine learning in which smart agents interact with their environment and take actions that maximize their cumulative reward. In 2015, when Alpha Go surpasses the human expert player [57], then reinforcement learning particularly deep reinforcement learning gained a lot of interest from academics, business, and commercial industries [58], [60]. Furthermore, reinforcement learning is a method for comprehending and carrying out goal-directed learning as well as producing optimum decisions. In addition, the algorithms of RL are applicable in application areas where data is complex and few to train with traditional deep learning models. More precisely, in general, at every time step T, an agent is presented with the situation of the environment in form of state s. After this perception of the environment, the agent performs the action a, for which it is awarded reward r and transition is done to move to the next state s'. The agent will be rewarded for correct behaviors and penalized for incorrect actions. Unlike human involvement, the agent learns by maximizing its rewards and decreasing its penalties. This entire process is defined as a Markov Decision Process (MDP) which is a set comprising of $< S, A, P, R, \gamma >$ in which S is a collection of finite states, A is referred to as a collection of actions, the probability of

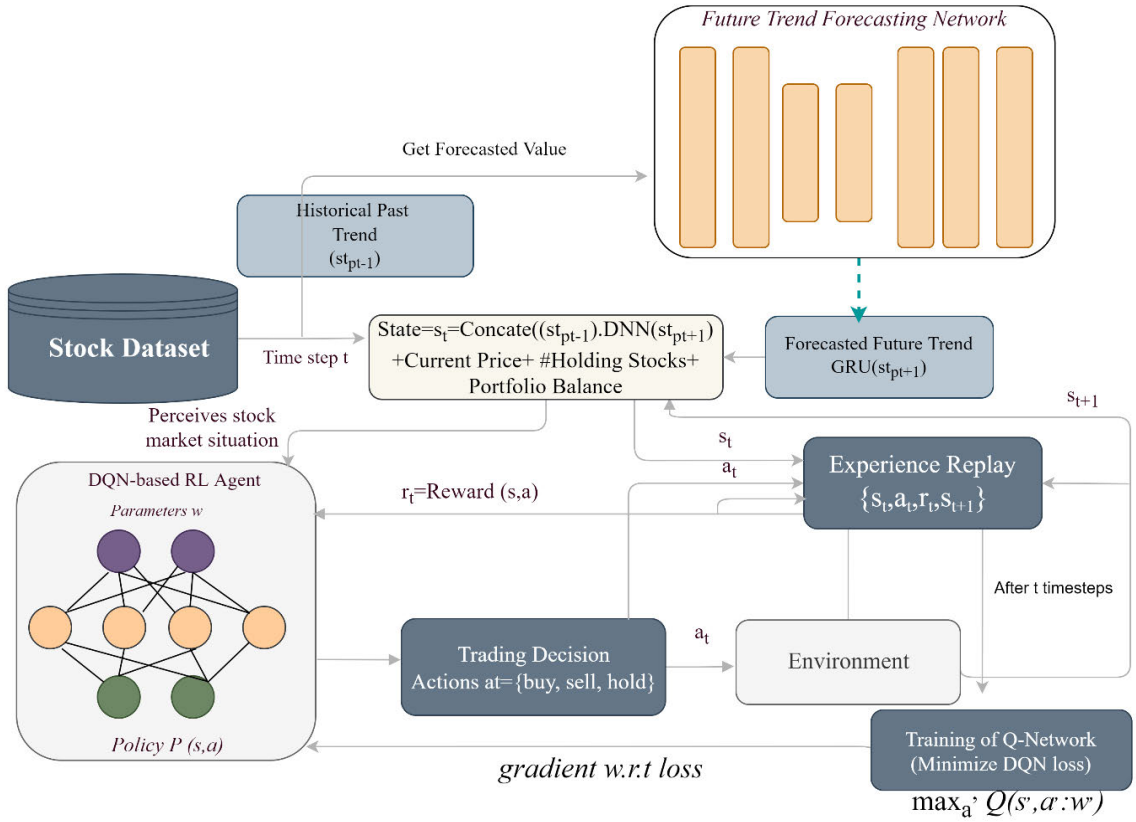


FIGURE 3. Pictorial overview of proposed methodology.

transition is defined as $P(s, a, s')$, and reward at any state s is denoted $R(s, a)$. Another important parameter is a discount factor γ having a value $\gamma \in \{0, 1\}$ that indicates the trade-off among the long-term rewards and immediate rewards.

Generally, in reinforcement learning for every kind of problem the states, rewards, and environments are formulated according to the design strategy and needed objective. For instance, in algorithmic trading, the state is not immediately given in and must be generated from a sequence of observations. To account for this, the MDP model was modified to involve an observation likelihood $P(o|s, a)$. The partly noticeable MDP (POMDP) framework is the name given to this extended model [61]. Moreover, the action at any state s is taken by an agent using either deterministic $\mu(s)$ or stochastic policy $\pi(a|s)$ in which for every state the probability distribution is defined for every possible action. At state s_t, the total discounted sum of future gains acquired by the agent is denoted as the discounted return G_t given in equation (1):

$$G_t = \sum_{i=0}^{\infty} \gamma^i_{r_{t+i}} = r_t + \gamma G_{t+1} \tag{1}$$

The mapping among the actions of states is accomplished through policy. In every state, a policy π is established to outline the action to be taken by an agent. During an agent's lifespan, its primary goal is to find an optimum policy that maximizes the expected total reward. The optimal policy π^* is specified by equation (2):

$$\pi^*(s) = arg\max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s', a) V^*(s', a) \tag{2}$$

For every pair of state-action $V^{\pi}(s, a)$ is a value function that is formalized.

This is an approximation of the intended reward as a set of policies. The maximum reward attained by the agent from different states is employed to determine the optimum policy which leads to providing the optimum value function. Equation (3) represents the best value function.

$$V^*(s, a) = R(s, a) + {}^{max}_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s', a) V^*(s', a) \tag{3}$$

As a result, the reinforcement learning agent learns from its surroundings through several encounters. There are several kinds of reinforcement learning algorithms. One of the simplest kinds of reinforcement learning algorithms is Q-learning. Q-Learning is a mechanism for determining which action an agent should do based on an action-value relation. This algorithm is one of the major advances in the RL paradigm by building the off-policy temporal difference scheme. By considering the targeted policy a state-action value function is assessed to determine the most valuable

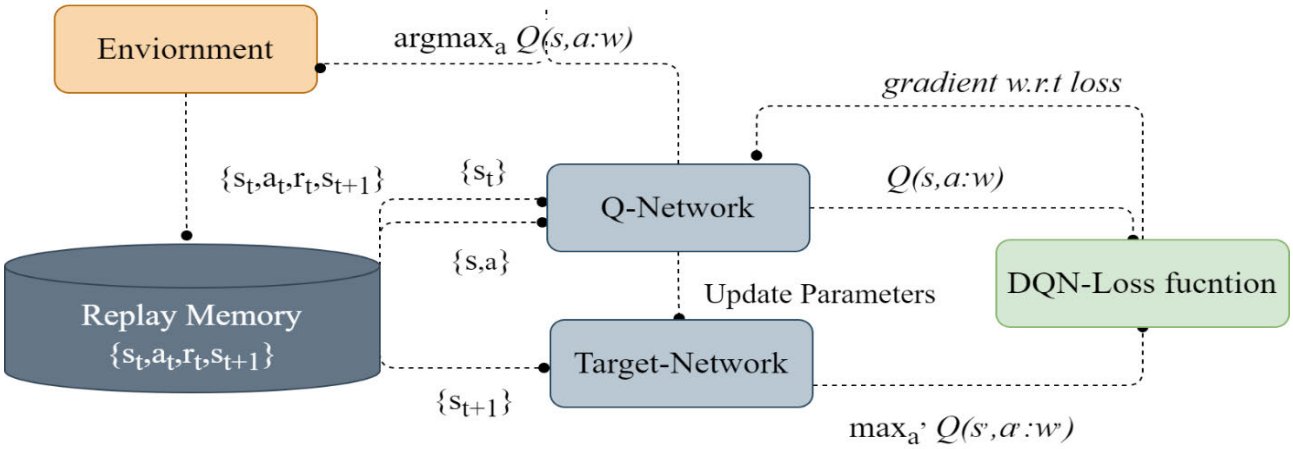


FIGURE 4. Experience replay memory is used to train the agents.

action. The whole process is formulated in equation (4):

$$Q_\pi\ (s,a) = \mathbb{E}\,|R_t|\,s_t = s, a_t = a, \pi\,| \tag{4}$$

In the above equation (4), the policy is indicated by π while the total reward earned by the agent is indicated by R_t computed using equation (5):

$$R_t = \sum_t \gamma^{t'-t_{r_{t'}}} \tag{5}$$

wherein 1.0 is a discount parameter for future rewards. The values in Q-table are updated in every episode using the Bellman equation given in equation (6):

$$Q^*\ (s,a) = r_t + \gamma_{a'}^{max} Q^*(s',a') \tag{6}$$

However, the problem with the Q-learning algorithm is the slow convergence and it is not suitable for problems having a large number of states. For such problems, advanced variants of deep reinforcement learning algorithms are employed.

B. DEEP-Q NETWORK FOR AUTOMATED STOCK TRADING

Deep Q-networks is a sophisticated reinforcement learning agent that maps the relationships between states and actions using Deep Neural Networks (DNN), which is equivalent to a Q-Table in Q-Learning. These DNNs can be any kind of neural network such as CNN, RNN, LSTM, etc. that can learn semantics from the raw data. Similar to Q-learning, the agent based on DNN can observe the sequence of states from the environment and performs an action over them and attain a reward depending upon an action. The weights of the DNN-based agent are updated by utilizing the Bellman equation. More precisely, the Q-values are produced in response to every action taken by an agent on states. Further, the primary goal of DNN is to learn and adjust its parameters depending on the rewards and penalties received. At the time of prediction, the trained DNN model predicts the optimum action from the collection of action space $a \in A$.

Moreover, in the collections of states, there are a lot of correlations that result in increasing the instability of the original Q-learning algorithm [62]. A very little alteration in the Q-value will lead to a substantial change in agent policy as well as the correlation between the target and Q-value. All these limitations are overcome in the Deep-Q networks by two different techniques i-e experience replay and repetitive updates as shown in Figure 4. Repetitively considering the Q-values in successive updates will minimize the correlations between the target and value. Similarly, the technique of experience replay will address the problem of correlation by smoothing the alterations in the data with the help of data randomness. Due to all these characteristics of a Deep-Q Network, it is commonly used for different problems such as in recommender systems [63], forecasting problems [32], and other robotics-related tasks [64]. In this study, we employed the Deep-Q networks for stock market trading, we trained a deep-reinforcement learning agent that assists the investors in when to buy, hold or sell the stocks to gain more profit. Generally, in RL algorithms e-g Deep-Q network, state engineering is a very challenging task. The more accurate the observation from the environment, the more optimum actions are taken by the agent. The step-by-step formulation aligned with this specific problem of stock trading of the proposed algorithm is described below:

1) STATES ENGINEERING

In order to assist the agent to learn the most optimum policy, an effective state representation of the environment has a significant impact. For the stock market, the environment for the agent is the current situation of the stock market. As a result, choosing a collection of data inputs is a requirement for trading agents to comprehend the stock market and develop trading rules. Catching market conditions at a certain moment is the fundamental factor of stock market trading. In existing studies, the historical trends of the stock prices i-e closing prices, or technical and fundamental indicators are employed as a state to make the agent observe the stock market. Since, the nature of the stock market is very volatile so taking the

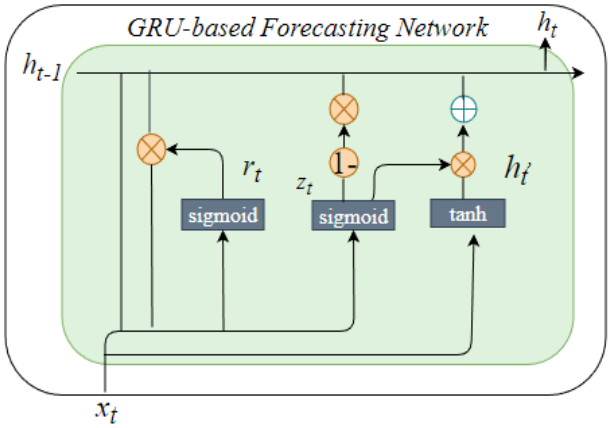


FIGURE 5. Simple architecture of the Gated Recurrent Unit.

trading action of buying, selling, or holding by only observing the stock prices at previous time steps $t-1$ is not enough. The trend of future stock prices $t+1$ should also be included to be observable by the agent. As a result, in this work, we presented an agent that is aware of both future and historical stock market circumstances. Therefore, the state of an agent is described in equation (7):

$$State\ (t) = s_t = concat(st_{p_{t-1}}.DNN(st_{p_{t+1}})) \tag{7}$$

where $st_{p_{t-1}}$ is the stock prices at previous time steps of days while the $st_{p_{t+1}}$ is the stock prices at future time steps. This $st_{p_{t+1}}$ is computed using a DNN-based model referred to as a forecasting network. In addition, $st_{p_{t-1}}$ and $DNN(st_{p_{t+1}})$ is a difference between the terms of the current stock price with past stock prices.

a: FORECASTING NETWORK BASED ON GRU

Gated Recurrent Unit (GRU) is one of the variants of recurrent neural networks (RNN) [65] as shown in Figure 5. It is less complex and faster than LSTM. The Gated Recurrent Unit, or GRU, has a similar methodology as the RNN, however, the distinction is in the functioning and gates connected with every GRU unit. GRU cells, like LSTM cells, do not have access to a distinct memory cell. The GRU is comprised of only two gates i-e update gate Z_t and reset gate r_t The quantum of past information memories is computed by GRU which is then saved for the usage in future. The mathematical expression of update and reset gates of GRU are given below in equations (8) to (11):

$$Z_t = \sigma(W_z h_{t-1} + W_z x_t + b_z) \tag{8}$$

$$r_t = \sigma(W_r h_{t-1} + W_r x_t + b_r) \tag{9}$$

$$h_{t'} = \tanh(W_r\ (h_{t-1} r_t) + W_f x_t + b_h \tag{10}$$

$$h_t = (1 - z_t)\, h_{t-1} + Z_t h_t' \tag{11}$$

In the above equations (8) to (11), at the time step t the input and output vector is indicated by x_t and h_t, while an array of weights is denoted by W with bias value b and σ denotes the sigmoid activation. In this research study, the GRU is employed as a forecasting model to get the one component of the state i-e. $st_{p_{t+1}}$. The architecture of the proposed forecasting network consists of GRU cells having neurons with tanh activations. In the last, dense layers are added having only one neuron to predict the closing price of the stocks. The model is trained with mean_squrared_error (MSE) as a loss function with the "Adam" optimizer. This forecasting network assists the reinforcement learning agent in taking the best decisions by providing it with the future situation of the stock markets.

2) RL AGENT

The reinforcement learning agent in the proposed Deep-Q network is based on deep neural networks as shown in Figure 5. Generally, the Deep-Q network has two networks, one is the main network and the other is the target network. These two networks have the same architecture based on the deep neural network but have different parameters and weights. After every N iteration, the main model's weights are transferred to the target model. This will result in more stable and effective learning. In the proposed work, the architecture of the RL agent consists of three dense layers with a number of units 24, 12, and 8 followed by a dropout layer and a final dense layer with three units indicating three types of actions namely, buy, sell, and hold. The last layer has a number of neurons equal to trading decisions i-e three (buy, sell and hold). Moreover, the optimizer is Adam with a learning rate of 0.0001, and the loss function is Huber. The RL agent makes trading decisions and learns through rewards and penalties.

3) ACTION SPACE

In the proposed work, the action space consists of three actions formulated as 1, 0, −1 or {buy, hold, sell}. The agent spends the money on buying the stocks when the action value is 1. Similarly, when it is 0, the agent will do nothing i-e it doesn't purchase or sell stocks on the stock exchange and when it is -1 the agent will sell the stocks in a way to earn more profit. Buying and selling phases are not always profitable.

4) REWARDS AND GOALS OF AGENT

When the agent performs an action, a reward in the form of numerical points is provided to the agent to indicate how good the action is at a given stage. For this problem, the reward in terms of profit is assigned to an agent. I-e. when the agent takes an action to sell the stocks then subtracting the price of buying from selling, provides the total profit generated by the trade action. The agent's primary goal is to maximize this profit. The larger the profit, the better the trading action. More precisely, the reward is given to the agent equal to the total profit generated when it sells stocks otherwise the reward is zero when a loss has occurred. The zero-value act as a penalty to the RL agent for doing the wrong action of making a loss during trading.

5) EXPERIENCE REPLAYS

In Deep-Q networks, the experience replays are the major part in which the agent's past history of observing states, actions,

and obtained rewards are saved. The performance of the agent is enhanced by picking the different samples of experience from the memory to train it. The Deep-Q-Network utilizes the DNN-based model that serves as a function optimization with weights θ. In the i^{th} iteration, the parameters or weights of the Q-Network are updated by minimizing the MSE loss using the Bellman equation. Among the target Q and predicted Q, the difference is computed through the loss function which is defined in equation (12):

$$Loss = \left(\left(r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta)\right)\right)^2 \quad (12)$$

In the above equation, this loss is minimized for the weights using the Adam optimizer. In addition, we train the deep Q-Network with the Huber loss function. During training, the Huber loss manages the stability of the algorithm. On the other hand, the MSE loss unfairly penalizes huge errors. However, the DNN predicts the values depending on its own input. Therefore, the MSE loss function has a substantial negative impact on the DQN algorithm. The weight updating in DQN is slower and needs more stability. Similarly, the MAE loss function is not differentiable at 0. So, the best option to train the DQN is the Huber loss which is a good trade-off among them, and it is defined in equation (13):

$$H(x) = \begin{cases} \frac{1}{2}x^2 & if\ |x| \leq 1 \\ |x| - \frac{1}{2} & otherwise \end{cases} \quad (13)$$

IV. EXPERIMENTS AND RESULTS

In this section, we evaluate the performance of our proposed model through different experimental settings. The performance of the trading agent is evaluated in terms of profit made during trading however the performance of the forecasting network is evaluated using root mean square errors, mean absolute errors, and mean squared error.

A. DATASET COLLECTION

In this research study, we have collected data from ten different markets namely Tesla, IBM, Amazon, US, Commodity Index Fund, SSE Composite Index, NIFTY50 Index, Chinese stock with code 601988 (with two different time spans), and CSCO stock (with two different time spans). The data for Tesla, IBM, Amazon, US, Commodity Index Fund, SSE Composite Index, and NIFTY50 Index was gathered from Yahoo Finance between January 1, 2017, and January 1, 2022. However, as with previous studies, we did the experimentation with the stock data employed in existing studies [66], [67]. For example, Chinese stocks and CSCO stocks with varying time durations, i.e. (2008-2018, 2005-2017, and 2000-2010). It was found that the chosen range of Tesla, IBM, Amazon, US, Commodity Index Fund, SSE Composite Index, and NIFTY50 Index also included the COVID-19 event, so it is more interesting to train a trading bot on stock data that includes such events, which may cause fluctuations in stock prices. The trading period of all ten stocks is depicted in Figure 6. It is observed from Figure 6, that Tesla has an upward trend, While IBM has more volatile patterns, it sometimes drops and sometimes rises in its value. Similarly, Tesla maintains a rising trajectory; however, the trends are fluctuating from 2019 to 2021 because of the influence of COVID. Moreover, as seen in Figure 6 part (h), the Chinese stock 601988's prices rose abruptly and subsequently fell. Likewise, U.S Commodity Index and NIFTY 50 have a sudden rise and drop trend in prices.

B. RESULTS OF STOCK FORECASTING NETWORK

The proposed stock forecasting network is based on the GRU model trained on the different stock datasets namely Tesla, IBM, Amazon, SSE Composite Index, US Commodity Index, NIFTY 50 Index, Chinese stock with code 601988 (with two different time spans), and CSCO stock (with two different time spans). Initially, the data of Tesla, IBM, Amazon, US, Commodity Index Fund, SSE Composite Index, and NIFTY50 Index is divided into train and test with a 60/40 ratio. However, Chinese stock with code 601988 and CSCO stock is divided into train and tests according to the division specified in the research studies [66], [67]. Following on, the models are trained on the training data, and later on, validation is performed over the test data. The results of the GRU-based forecasting model in terms of actual and predicted close prices for both train and test are depicted in Figure 7. In Figure 7, the first graph is the result of tesla stock. The x-axis of the graphs denotes the time span in form of days over 5 years while the y-axis denotes the closing price. The red lines in the graphs indicate the predictions of the GRU model over the training data while the purple lines indicate the predictions of the GRU model over the test data. The black curves in the graph show the actual values of the dataset. It is observed from the all graphs given in Figure 7 that the GRU model shows the best results with the IBM, U.S Commodity Index, SSE Composite Index, Chinese stock 601988 and CSCO stock, however, with the historical data of Tesla, NIFTY50, and Amazon, the performance of the GRU model drops at the ending day's i-e 2020 to 2022 period. All of these graphs show the result of the GRU model with the loopback variable set to 1. More precisely, we consider only one previous time step t to forecast the future time step. On the contrary, we have also performed experiments with greater window sizes or previous time steps.

More precisely, the results of loopback with 1, 2, and 3 in terms of RMSE, MSE, and MAE are given in Table 1. Each row of Table 1 shows the results of a particular stock with different window sizes. The value of RMSE, MSE, and MAE for tesla stocks in the case of window size 1 is 0.05, 0.003, and 0.04 respectively. Similarly, the RMSE, MSE, and MAE for IBM stocks are 0.02, 0.0006, and 0.01 while for Amazon it is 0.08, 0.007, and 0.7. Following on, with SSE composite Index the values of RMSE, MSE, and MAE are about 0.028, 0.0008, and 0.0220, while for the U.S Commodity index it is 0.2071, 0.00042, and 0.01489, and similarly, for NIFTY these values are 0.0398, 0.00158, and 0.02873 respectively. In addition, on Chinese and CSCO stocks, the error rate is

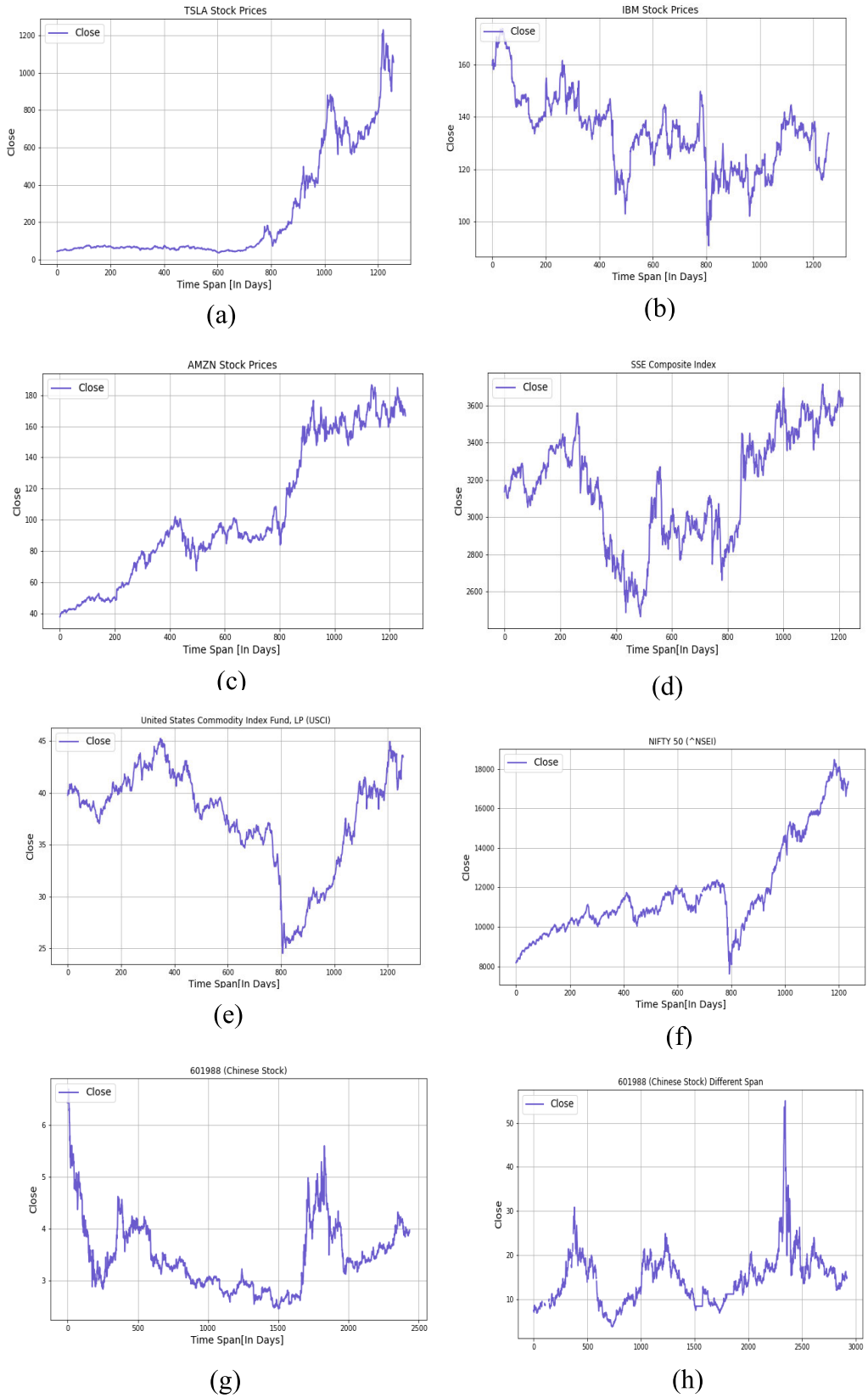


FIGURE 6. Historical trends of different stock markets including (a) TSLA (b) IBM (c) Amazon (d) SSE Composite Index (e) US Commodity Index Fund, and (f) NIFTY 50 Index (g) Chinese Stock with code 600198 (2008-2018) (h) Chinese Stock with code 600198 (2005-2017) (i) CSCO Stock (2008-2018) (j) CSCO Stock (2002-2010).

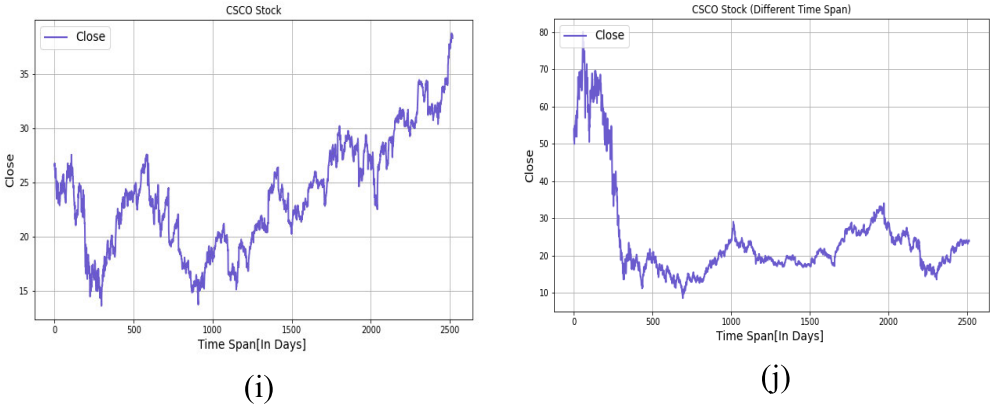


FIGURE 6. *(Continued.)* **Historical trends of different stock markets including (a) TSLA (b) IBM (c) Amazon (d) SSE Composite Index (e) US Commodity Index Fund, and (f) NIFTY 50 Index (g) Chinese Stock with code 600198 (2008-2018) (h) Chinese Stock with code 600198 (2005-2017) (i) CSCO Stock (2008-2018) (j) CSCO Stock (2002-2010).**

TABLE 1. **Results of GRU-based Single-Time step Forecasting Network with different loop back values.**

Single Time Steps Forecasting					
Stock Name	**RMSE**	**MSE**	**MAE**	**Loopback**	**Train/Test Ratio**
Tesla	0.05816753	0.0033834614	0.041334692	1	800/460
IBM	0.025732385	0.0006621557	0.018003164	1	800/460
Amazon	0.084359914	0.0071165953	0.07727872	1	800/460
SSE Composite Index	0.02878	0.0008286101	0.0220964	1	800/460
US commodity Index Fund, LP	0.020701	0.00042854	0.01498	1	800/460
NIFTY 50	0.03985	0.001588	0.02873	1	800/460
601988 Chinese Stock	0.00861	0.005771	0.000741	1	1948/488
601988 Chinese Stock	0.0108	0.00011	0.00853	1	1948/488
CSCO Stock	0.01913	0.00036	0.01378	1	2016/503
CSCO Stock	0.00769	0.00005922	0.005900	1	2016/503
Tesla	0.12319026	0.01517584	0.09140169	2	800/460
IBM	0.027494555	0.0007559506	0.018945761	2	800/460
Amazon	0.046330802	0.0021465432	0.040985994	2	800/460
SSE Composite Index	0.02915	0.00085025	0.0224332	2	800/460
US commodity Index Fund, LP	0.01707	0.000549	0.01707	2	800/460
NIFTY 50	0.029467	0.0008683	0.022816	2	800/460
601988 Chinese Stock	0.0100	0.000100	0.007	2	1948/488
601988 Chinese Stock	0.01218	0.00014	0.0101	2	1948/488
CSCO Stock	0.0247	0.00061	0.01885	2	2016/503
CSCO Stock	0.0074	0.00005868	0.0056	2	2016/503
Tesla	0.055375442	0.0030664396	0.038218603	3	800/460
IBM	0.026604127	0.0007077796	0.019127147	3	800/460
Amazon	0.10468853	0.010959689	0.09619159	3	800/460
SSE Composite Index	0.000844	0.00084	0.02240	3	800/460
US commodity Index Fund, LP	0.02036	0.00041	0.0160	3	800/460
NIFTY 50	0.05919	0.00350	0.0414	3	800/460
601988 Chinese Stock	0.0109	0.0001	0.00864	3	1948/488
601988 Chinese Stock	0.0088	0.0000786	0.006079	3	1948/488
CSCO Stock	0.02784	0.000775	0.02048	3	2016/503
CSCO Stock	0.0073	0.0000545	0.00555	3	2016/503

also less. It is observed from these results that the GRU model show very effective and best results for forecasting problem. Following on, the size of the window i-e number of previous time steps to forecast future time steps is also studied by setting different values of the loopback variable. More precisely, the RMSE, MSE, and MAE scores of Tesla stock for the loopback variable set to 2 are 0.12, 0.01, and 0.09. It is observed that the value of RMSE, as well as MSE and MAE, becomes higher when we consider more previous time steps. This behavior is also observed with the other stocks.

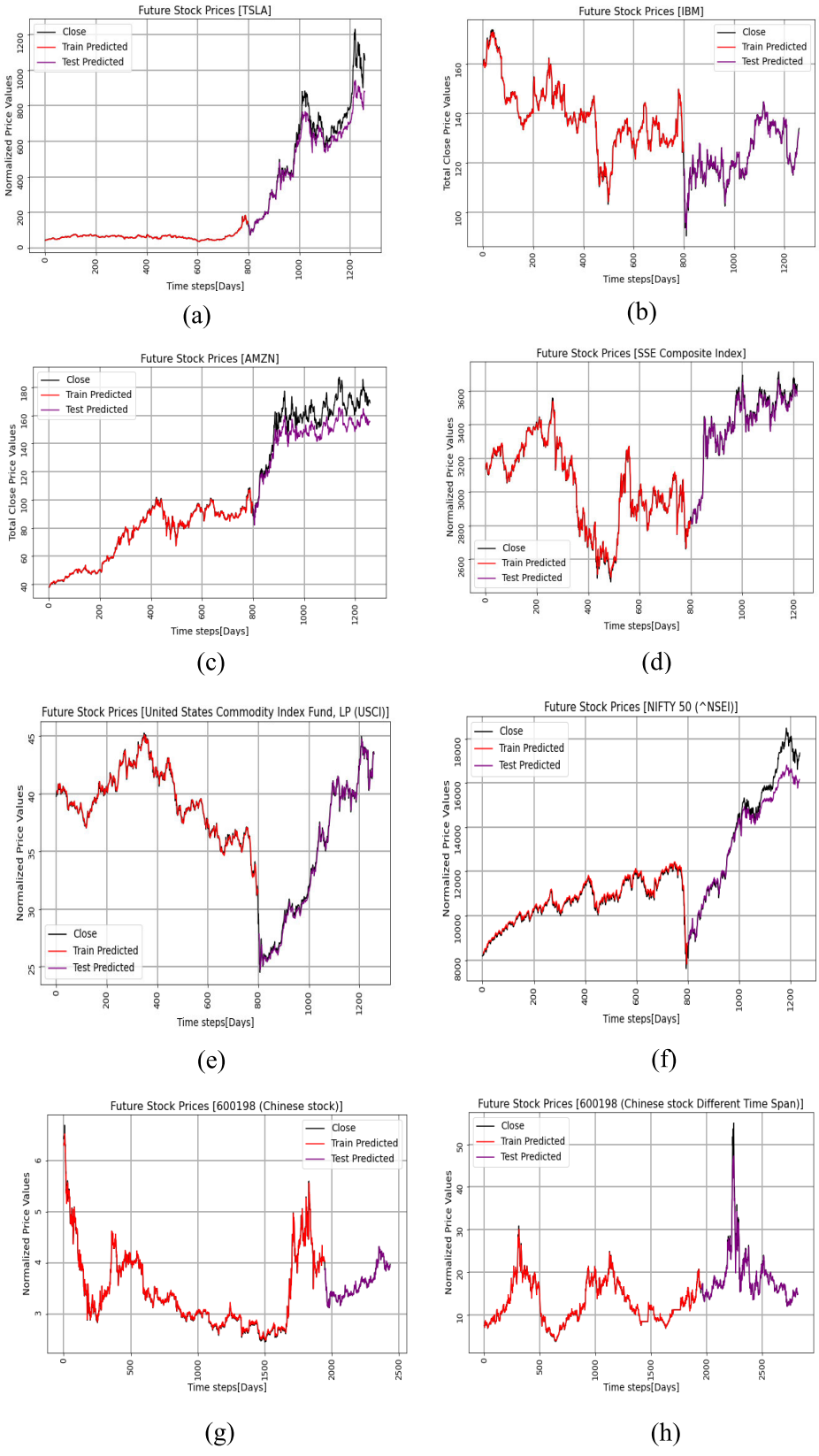


FIGURE 7. Performance of GRU-based forecasting network over different stock markets (a) TSLA Stock market future trends (b) IBM Stock market future trends (c) Amazon Stock market future trend (d) SSE Composite Index future trend (e) US Commodity Index Fund future trend (f) NIFTY 50 Index future trend (g) Chinese Stock with code 601988 (2008-2018) (h) Chinese Stock with code 601988 (2005-2017) (i) CSCO Stock (2008-2018) (j) CSCO Stock (2002-2010).

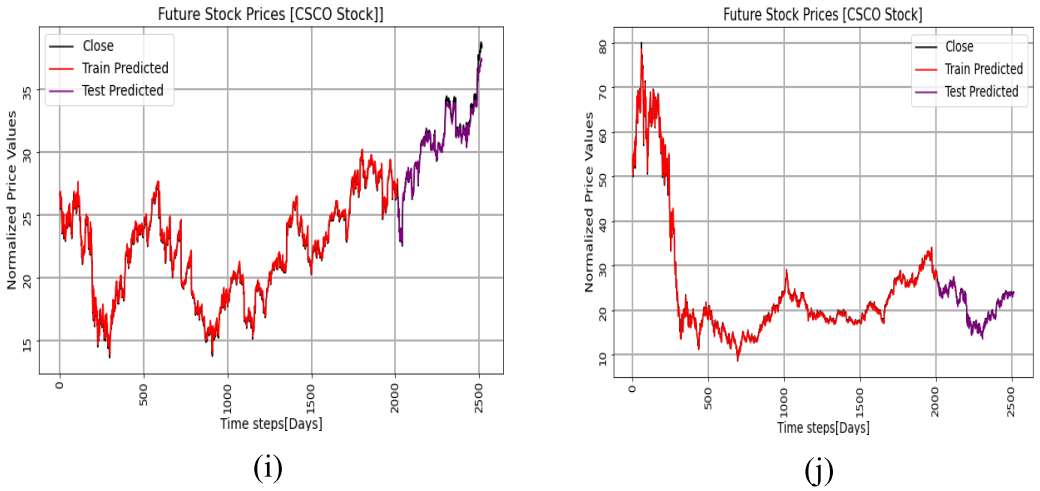


FIGURE 7. ***(Continued.)*** **Performance of GRU-based forecasting network over different stock markets (a) TSLA Stock market future trends (b) IBM Stock market future trends (c) Amazon Stock market future trend (d) SSE Composite Index future trend (e) US Commodity Index Fund future trend (f) NIFTY 50 Index future trend (g) Chinese Stock with code 601988 (2008-2018) (h) Chinese Stock with code 601988 (2005-2017) (i) CSCO Stock (2008-2018) (j) CSCO Stock (2002-2010).**

Similarly, if we consider the three-time steps, then the value of RMSE, MSE, and MAE is higher with Amazon stocks, however, it is slightly better in the other stocks. It is observed that in most cases if we increase the value of window size or previous time steps then the results drop instead of improving. However, with one previous time step as input, it shows very stable and best results. Another noteworthy thing that needs to be mentioned here is that the proposed forecasting network is based on GRU having only four units followed by a dense layer. This indicates that the forecasting network is not very complex and large in terms of trainable parameters but still produces the best results. It perfectly predicts future prices with the least errors. The parameters of the GRU model are minimal, giving it a lightweight model that allows for fast training.

Following on, we validated the performance of the proposed GRU model in multi-time step forecasting. In this scenario, we run trials with different past time steps by changing the value of the loopback variable to 1, 2, or 3 to anticipate the next two prices, resulting in a multi-time step prediction problem. The results of the proposed forecasting network with a multi-time steps scenario are given in Table 2. The model is validated on ten different stocks namely Tesla, IBM, Amazon, SSE composite index, U.S commodity index, NIFTY 50 Index, Chinese stock with code 601988 (with two different time spans), and CSCO stock (with two different time spans). It is observed that the forecasting model based on GRU is good enough to predict the next two future prices and as a result, it will be more useful in enabling RL agents to be informed of the following two days' future pricing before making trading decisions at any time step. Furthermore, if we increase the window size of previous time steps to be taken as input to GRU, then the results are also encouraging. As shown in Table 2 that if we set the value of the loopback variable to 2 then the GRU model accepts two previous time steps' prices to forecast the next two days' prices. In this case, the value of RMSE, MSE, and MAE for Tesla stocks are 0.05, 0.003, and 0.0 which is very less. The same behavior is observed with the remaining stocks. In addition, with the loopback variable of value 3 i-e with the window size of 3, the model forecasting is also optimal. The values of RMSE, MSE, and MAE for Tesla stocks are 0.05, 0.003, and 0.02, while for IBM the values are 0.04, 0.002, and 0.013, and for Amazon, the values are 0.07, 0.005, and 0.0517 respectively.

Similarly, for SSE Composite index these error values are 0.06255, 0.0039, and 0.0154, while for U.S composite index, these values are 0.0248, 0.000617, and 0.01879, and in the last the error values for NIFTY50 are about 0.0620, 0.00385, and 0.0255. In addition, the error values for Chinese stock are 0.00978, 0.0009574, and 0.0071 while CSCO stock has 0.0279, 0.0007827, and 0.02229 respectively. The proposed forecasting model is best suited to guide the reinforcement learning-based agent about the future stock price of the next day as its states so that it makes effective trading decisions.

C. RESULTS OF RL AGENT

In the second phase, we designed the stock market trading agent based on reinforcement learning that observes the market situation to take such actions in the trading environment that ultimately increase the profit for the company/business. The RL agent is simply the deep neural network having four dense layers. This network learns from the reward it gains over the different actions. To validate the performance of the RL agent, we perform experiments with different window sizes of historical prices. For offline testing, we train the model for 5000-time steps, and later on we use that trained model to make predictions on the test data of stocks of the different trading periods. In this case, while making decisions

TABLE 2. Results of GRU-based MULTI-Time step Forecasting Network with different loop back value.

Multi Time Steps Forecasting					
Stock Name	**RMSE**	**MSE**	**MAE**	**Loopback**	**Train/Test Ratio**
Tesla	0.078884837	0.0062228175	0.047158679	1	800/460
IBM	0.04630290	0.002143958	0.0133934	1	800/460
Amazon	0.06035032	0.0036421614	0.03110537	1	800/460
SSE Composite Index	0.05958	0.00355	0.01195	1	800/460
US commodity Index Fund, LP	0.02356	0.00055	0.0177	1	800/460
NIFTY 50	0.05533	0.003061	0.01389	1	800/460
601988 Chinese Stock	0.00978	0.0009574	0.0071	1	1948/488
601988 Chinese Stock	0.0290	0.00084	0.0155	1	1948/488
CSCO Stock	0.0279	0.0007827	0.02229	1	2016/503
CSCO Stock	0.00886	0.00007859	0.006726	1	2016/503
Tesla	0.05597076	0.003132726	0.02135611	2	800/460
IBM	0.049494483	0.0024497038	0.017912727	2	800/460
Amazon	0.067923264	0.004613569	0.04321921	2	800/460
SSE Composite Index	0.06177	0.0038164	0.014211	2	800/460
US commodity Index Fund, LP	0.02659	0.0007074	0.02115	2	800/460
NIFTY 50	0.05695	0.003244	0.020592	2	800/460
601988 Chinese Stock	0.0109	0.000120	0.0075	2	1948/488
601988 Chinese Stock	0.033	0.0011	0.0256	2	1948/488
CSCO	0.0306	0.0009377	0.02408	2	2016/503
CSCO	0.0088	0.00343774	0.0067	2	2016/503
601988 Chinese Stock	0.0288	0.000833	0.01569	3	1948/488
Tesla	0.05863232	0.00343774	0.0266366	3	800/460
IBM	0.04690839	0.002200397	0.0137756	3	800/460
Amazon	0.07381795	0.005449089	0.051714874	3	800/460
SSE Composite Index	0.062559	0.003913	0.01540	3	800/460
US commodity Index Fund, LP	0.0248	0.0006177	0.01879	3	800/460
NIFTY 50	0.062053	0.0038506	0.02558	3	800/460
601988 Chinese Stock	0.0288	0.000833	0.01569	3	1948/488
601988 Chinese Stock	0.0134	0.00018	0.0108	3	1948/488
CSCO Stock	0.02704	0.00073	0.01959	3	2016/503
CSCO Stock	0.00955	0.0000912	0.00731	3	2016/503

the model does not updates itself by taking wrong decisions i-e the model does not update its weights depending upon the experiences stored in the experience replay. With a window size of the previous 10 historical prices, we first train the RL agent on historical data of stock from 01 January 2017 to 01 January 2018. Later on, we test it over stock data from 01 January 2019 to 01 January 2022. These dates are fixed for Tesla, IBM, Amazon, US, Commodity Index Fund, SSE Composite Index, and NIFTY50 Index while for Chinese 600198 stock and CSCO stock, we have followed the existing approach [66], [67]. During training, at any instant time, the agent observes the differences between the current stock price with previous stock prices, the stocks available in the inventory, the total balance available, and the future stock price of the next day. From this available information, the agent is trained to take such trading decisions in which more profit is generated. Hence, the agent is awarded in terms of profit generated when it takes a good trading decision. After training the agent, it is put into the test mode and validated to make trading decisions. During testing, the agent initially has a balance of 50,000. The inventory is empty since the agent does not buy anything initially.

We trained separate agents for each of the stocks. The results of Tesla, IBM, Amazon, SSE composite index, U.S commodity Index fund, NIFTY 50, Chinese stock with code 601988 (with two different time spans), and CSCO stock (with two different time spans) are depicted in Figure 8. The graphs in Figure 8 show the results of trading agent by considering either one day's next future prices (single step) as an observation state and the next two days' future prices (multi-step). More exactly, the trading agent's states in the first case comprise just one day's worth of future price features. However, in multi-step, the agent is aware of the next two-day future prices before making the decision on current-day stocks. To accomplish this, we train the model i-e forecasting network to take previous time steps values of stocks to predict the values of the future horizon of two-time steps.

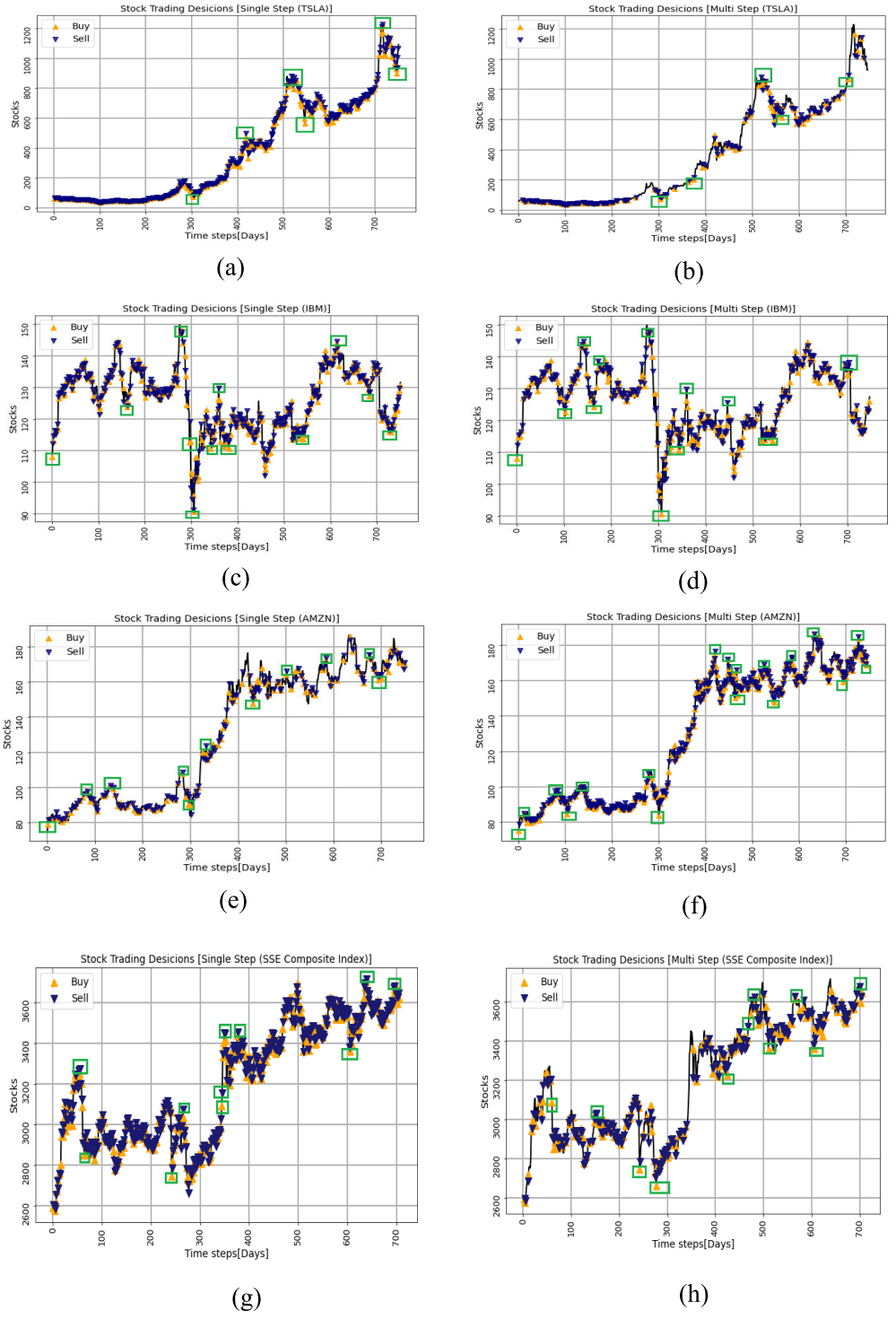


FIGURE 8. Performance of RL trading agent in terms of trading decisions over six different stock markets (a,b) Decisions on TSLA Stock market trends with both single and multi-step ahead features and window size of 10(c,d) Decisions on IBM Stock market future trends with both single and multi-step ahead features and window size of 10 (e,f) Decisions on Amazon Stock market future trend with both single and multi-step ahead features and window size of 10 (g,h) Decisions on SSE Composite Index future trend with both single and multi-step ahead features and window size of 10 (i,j) Decisions on US Commodity Index Fund future trend with both single and multi-step ahead features and window size of 10(k,l) Decisions on NIFTY 50 Index future trend with both single and multi-step ahead features and window size of 10 (m,n) Decisions on Chinese 601988 stock (2018-2018) future trend with both single and multi-step ahead features and window size of 10 (o,p) Decisions on Chinese 601988 stock(2015-2017) future trend with both single and multi-step ahead features and window size of 10 (q,r) Decisions on CSCO stock (2018-2018) future trend with both single and multi-step ahead features and window size of 10 (s,t) Decisions on CSCO stock(2000-2010) future trend with both single and multi-step ahead features and window size of 10.

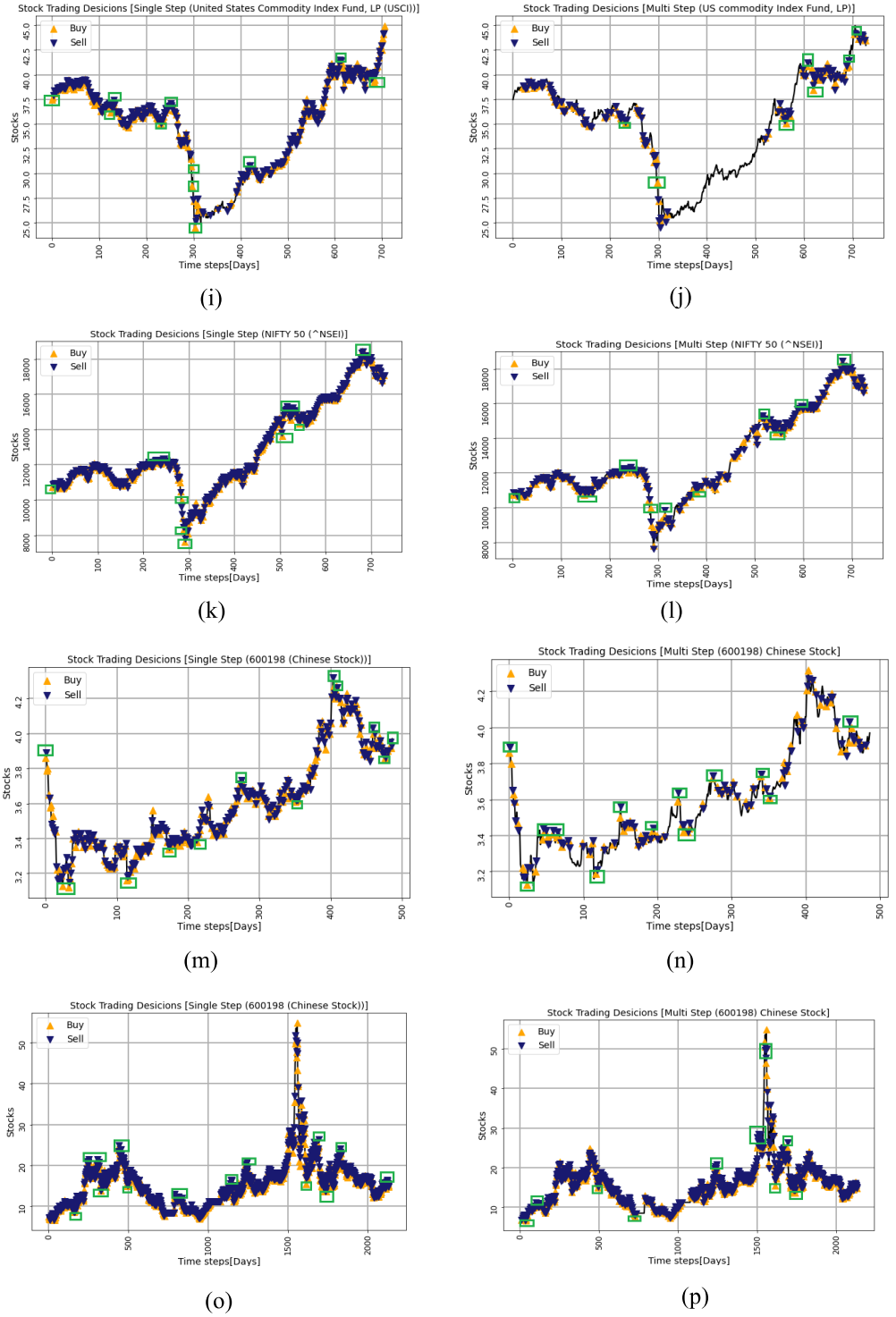


FIGURE 8. *(Continued.)* **Performance of RL trading agent in terms of trading decisions over six different stock markets (a,b) Decisions on TSLA Stock market trends with both single and multi-step ahead features and window size of 10(c,d) Decisions on IBM Stock market future trends with both single and multi-step ahead features and window size of 10 (e,f) Decisions on Amazon Stock market future trend with both single and multi-step ahead features and window size of 10 (g,h) Decisions on SSE Composite Index future trend with both single and multi-step ahead features and window size of 10 (i,j) Decisions on US Commodity Index Fund future trend with both single and multi-step ahead features and window size of 10(k,l) Decisions on NIFTY 50 Index future trend with both single and multi-step ahead features and window size of 10 (m,n) Decisions on Chinese 601988 stock (2018-2018) future trend with both single and multi-step ahead features and window size of 10 (o,p) Decisions on Chinese 601988 stock(2015-2017) future trend with both single and multi-step ahead features and window size of 10 (q,r) Decisions on CSCO stock (2018-2018) future trend with both single and multi-step ahead features and window size of 10 (s,t) Decisions on CSCO stock(2000-2010) future trend with both single and multi-step ahead features and window size of 10.**

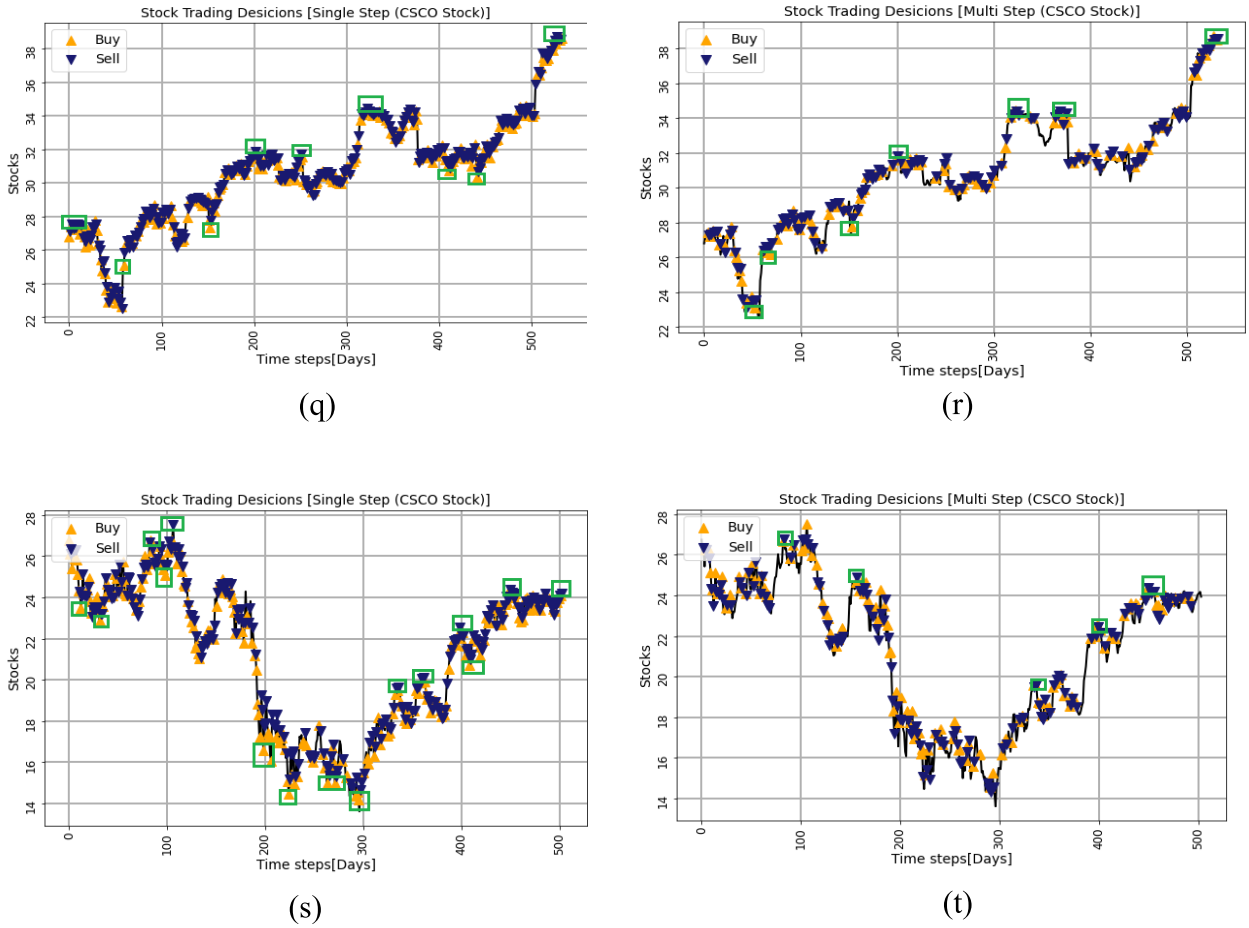


FIGURE 8. ***(Continued.)*** **Performance of RL trading agent in terms of trading decisions over six different stock markets (a,b) Decisions on TSLA Stock market trends with both single and multi-step ahead features and window size of 10(c,d) Decisions on IBM Stock market future trends with both single and multi-step ahead features and window size of 10 (e,f) Decisions on Amazon Stock market future trend with both single and multi-step ahead features and window size of 10 (g,h) Decisions on SSE Composite Index future trend with both single and multi-step ahead features and window size of 10 (i,j) Decisions on US Commodity Index Fund future trend with both single and multi-step ahead features and window size of 10(k,l) Decisions on NIFTY 50 Index future trend with both single and multi-step ahead features and window size of 10 (m,n) Decisions on Chinese 601988 stock (2018-2018) future trend with both single and multi-step ahead features and window size of 10 (o,p) Decisions on Chinese 601988 stock(2015-2017) future trend with both single and multi-step ahead features and window size of 10 (q,r) Decisions on CSCO stock (2018-2018) future trend with both single and multi-step ahead features and window size of 10 (s,t) Decisions on CSCO stock(2000-2010) future trend with both single and multi-step ahead features and window size of 10.**

The black curves in the graphs show the stock price data, while the yellow points indicate that at this time step the agent takes the buy decision, while the blue points indicate that at this time step the agent takes the sell decision. However, at some time steps over the trading period the agent does not takes any decision i-e Hold. By observing the graph of Tesla stock when the stock values are high the agent takes the sell decision to earn more profit while when the values of the stocks are low the agent takes the buy decision. Some best decisions are highlighted in form of green boxes to indicate the effectiveness of the suggested agent. Moreover, the results of IBM stock and SSE composite Index which are more volatile in nature having more fluctuating trends in prices are also perfectly handled by the proposed trading agent. At the extremes of low values of stocks, the model takes buy which is good as usually the optimal trading is the one in which we buy at low prices and sell at high prices to gain more profit. Similarly, the graphs of Amazon, NIFTY 50, U.S commodity, Chinese 601988 stock, and CSCO stock also shows the best performance.

It is observed from the results that the proposed trading agent is effective and best to work as a decision support system for automated trading to support the investors, decisions, and policymakers of different companies The average profit over the completed trading period (i-e on each day in the range 01 January 2019 to 01 January 2022), the total profit over the complete trading period, and portfolio value after the trading period is given in Table 3 and Table 4 for both single and multi-step ahead. It is observed from Table 3 that on average the model earns more profit with single-step features. In all of the experiments in Figure 8, the window size of previous historical prices that are taken into account while constructing states is 10. In addition, DQN (deep-Q learning) method, like many DRL algorithms, has

TABLE 3. Results of RL agent in terms of profits and portfolio values over single time steps with window size 10.

Stock Name	Portfolio value	Total profit over trading profit[1]	Average Profit over trading period[1]	Loopback	Time step
Tesla	1981.32	8515.623	11.339046	1	Single
Tesla	700.16	5348.883	7.122	2	Single
Tesla	831	5721.60	7.61	3	Single
Average	1170.82	6528.702	8.690	-	Single
IBM	-0.82	613.37	0.81	1	Single
IBM	71.61	712.16	0.94	2	Single
IBM	50.55	765.94	1.01	3	Single
Average	40.44	697.15	0.92	-	Single
Amazon	23.78	494.01	0.65	1	Single
Amazon	92.54	848.31	1.12	2	Single
Amazon	21.49	962.66	1.28	3	Single
Average	45.93	768.32	1.016	-	Single
SSE Composite Index	434.00	9269.80	13.111	1	Single
SSE Composite Index	434.31	9270	13.3	2	Single
SSE Composite Index	434.31	9270	13.2	3	Single
Average	434.20	9269.933	13.20	-	Single
US commodity Index Fund, LP	12.28	255	0.36	1	Single
US commodity Index Fund, LP	12.81	146	0.2068	2	Single
US commodity Index Fund, LP	-2.48	104	0.13	3	Single
Average	7.5366	168.333	0.232266	-	Single
NIFTY 50	283.74	40095.8	54.925	1	Single
NIFTY 50	384.64	39911.29	54.974230	2	Single
NIFTY 50	384.64	39911.290	54.974	3	Single
Average	351.00	39972.79	54.957	-	Single
601988 (Stock-2008-2018)	0.17	14.58	0.03	1	Single
601988 (Stock-2008-2018)	-0.07	5.98	0.01	2	Single
601988 (Stock-2008-2018)	-0.19	5.98	0.011	3	Single
Average	-0.03	8.846667	0.017	-	Single
601988 (Stock-2005-2017)	-73.68	479.87	0.22	1	Single
601988 (Stock-2005-2017)	-33.83	466.92	0.219	2	Single
601988 (Stock-2005-2017)	-33.28	504.5	0.23	3	Single
Average	-46.93	483.766	0.223	-	Single
CSCO(Stock-2008-2018)	17.62	112.12	0.21	1	Single
CSCO(Stock-2008-2018)	1.31	71.799	0.134	2	Single
CSCO(Stock-2008-2018)	3.29	84.20	0.15	3	Single
Average	7.40667	89.373	0.1644	-	Single
CSCO(Stock-2000-2010)	5.97	113.13	0.224	1	Single
CSCO(Stock-2000-2010)	10.45	122.12	0.242	2	Single
CSCO(Stock-2000-2010)	20.27	152.47	0.30	3	Single
Average	12.23	129.24	0.255	-	Single

[1] *This profit does not include loss during trading at different trading positions*

a significant amount of variance. Multiple training attempts with the same beginning conditions will invariably result in somewhat different trading strategies with various results. This is also observed in the research study of Théate and Ernst [68] and this is also discussed by Jia et al. [69]. Hence it is necessary to validate the model with multiple trials and report the average. One reason for this behavior might be the case that the agent starts learning initially by taking random actions. It is more likely that initially, e-g. the agent takes the random actions of more buy actions and experiences replay memory filled with more buy actions and fewer sell actions, and in this case, the positive reward given to the agent is less.

In such a case, the agent is less likely to learn the sell action that maximizes the objective i-e profit. Furthermore, we also validated the performance of the agent by giving the previous historical prices trend of larger window size i-e window sizes 15. The same set of experiments for large window sizes for both single and multi-step is shown in Figure 11. The black curves in the graphs represent stock price data, while the yellow dots indicate that the agent makes a purchase decision at this moment, and the blue points indicate that the agent makes a sell decision at this time. However, at various points throughout the trading period, the agent does not make a decision, i.e. hold. Observing the graph of Tesla stock, when the stock values are high, the agent sells to gain more profit,

TABLE 4. Results of RL agent in terms of profits and portfolio values over Multi-Time Steps with window size 10.

Stock Name	Portfolio value	Total profit over trading profit[1]	Average Profit over trading period[1]	Loopback	Time step
Tesla	189.09	4318.22	5.78	1	Multi
Tesla	-23.85	4838.27	6.48	2	Multi
Tesla	-10.35	6647.23	8.91	3	Multi
Average	51.63	5267.90	7.055	-	Multi
IBM	33.33	582.04	0.78	1	Multi
IBM	-0.36	647.31	0.86	2	Multi
IBM	31.23	600.6	0.47	3	Multi
Average	21.4	609.98	0.703	-	Multi
Amazon	-4.71	623.56	0.49	1	Multi
Amazon	6.25	657.03	0.88	2	Multi
Amazon	19.28	797.7	1.06	3	Multi
Average	6.94	692.76	0.81	-	Multi
SSE Composite Index	471.80	9651.94	13.651	1	Multi
SSE Composite Index	215.56	9815.04	13.88	2	Multi
SSE Composite Index	-524.37	8903.67	12.59	3	Multi
Average	54.33	9456.88	13.373	-	Multi
US commodity Index Fund, LP	-8.64	106.010	0.145	1	Multi
US commodity Index Fund, LP	-2.67	86.7499	0.1193	1	Multi
US commodity Index Fund, LP	-0.36	80.850	0.111	1	Multi
Average	-3.98	91.2033	0.1251	-	Multi
NIFTY 50	4105.25	99580.42	136.97444	1	Multi
NIFTY 50	1937.60	53713.0	73.88	3	Multi
NIFTY 50	-2680.65	45978.45	63.244	3	Multi
Average	1120.733	66423.80	91.366	-	Multi
601988 (Stock-2008-2018)	-0.19	1.24	0.025	1	Multi
601988 (Stock-2008-2018)	-0.44	5.25	0.01	2	Multi
601988 (Stock-2008-2018)	-1.99	2.1	0.0043	3	Multi
Average	-0.8733	2.8633	0.0131	-	Multi
601988 (Stock-2005-2017)	-59.26	520.24	0.244	1	Multi
601988 (Stock-2005-2017)	-7.33	406.61	0.199	2	Multi
601988 (Stock-2005-2017)	27.63	429.93	0.201	3	Multi
Average	-12.9867	452.26	0.21447	-	Multi
CSCO(Stock-2008-2018)	5.91	66.31	0.12	1	Multi
CSCO(Stock-2008-2018)	-5.50	48.58	0.09	2	Multi
CSCO(Stock-2008-2018)	6.09	121.09	0.22	3	Multi
Average	2.16667	78.666	0.1433	-	Multi
CSCO(Stock-2000-2010)	1.38	105.05	0.2088	1	Multi
CSCO(Stock-2000-2010)	2.27	148.27	0.294	2	Multi
CSCO(Stock-2000-2010)	-35.43	36.67	0.0729	3	Multi
Average	-10.5933	96.66	0.1919	-	Multi

[1] *This profit does not include loss during trading at different trading positions*

but when the stock values are low, the agent buys. Some of the better selections are highlighted in the form of green boxes to illustrate the efficacy of the proposed agent. Table 5 and Table 6 show the average profit over the completed trading period (i.e., on each day from January 1, 2019, to January 1, 2022), total profit over the trading period, and portfolio value after the trading period for both single and multi-step forward. Similar to the previous experiment, the agent takes the best decisions with the single-step ahead-based state rather than multi-step ahead.

To observe the increase in the portfolio, return over the trading period, Figure 9 shows the graphs for different stocks with different window sizes of previous time steps as well as single future price and multi-step ahead future price. It is observed that as the trading agent makes decisions, the portfolio return goes on increasing for all stocks. However, in some cases, the portfolio returns drop when the agent takes the wrong decisions. It is observed that the portfolio returns are high for Tesla stock and the NIFTY 50 Index rather than for the other stocks. The agent finds difficulty in earning a profit over IBM and SSE composite index due to its more volatile nature. Furthermore, the loss curves of deep Q Network during its training over the different stocks with different times or different window sizes of previous time steps are also depicted in Figure 10. The first eight graphs show the loss graphs of deep Q Network with window size 10 while the last eight graphs of Figure 10 show the results of all stocks namely Tesla, IBM, Amazon, SSE composite index, U.S

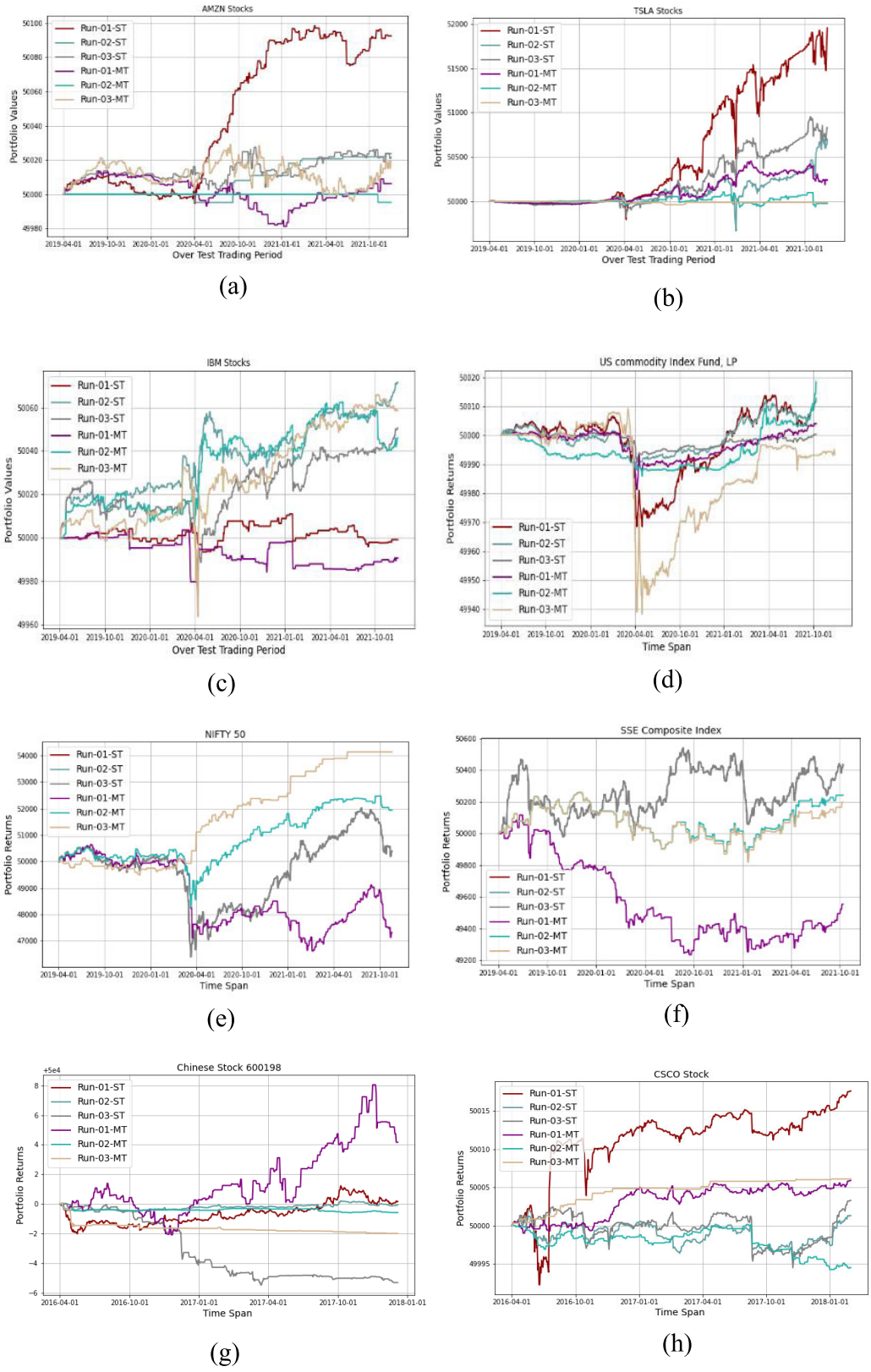


FIGURE 9. Performance of RL trading agent in terms of Portfolio values over ten different stock markets (a,b,c,d,e,f,g,h) Portfolio values on TSLA, IBM, AMZN, SSE Composite Index, US Commodity Index, NIFTY 50 Index trends Chinese and CSCO with both single and multi-step ahead features and window size of 10 (,i,j,k,l,m,n,o,p) Portfolio values TSLA, IBM, AMZN, SSE Composite Index, US Commodity Index, NIFTY 50 Index, Chinese and CSCO future trends with both single and multi-step ahead features and window size of 15.

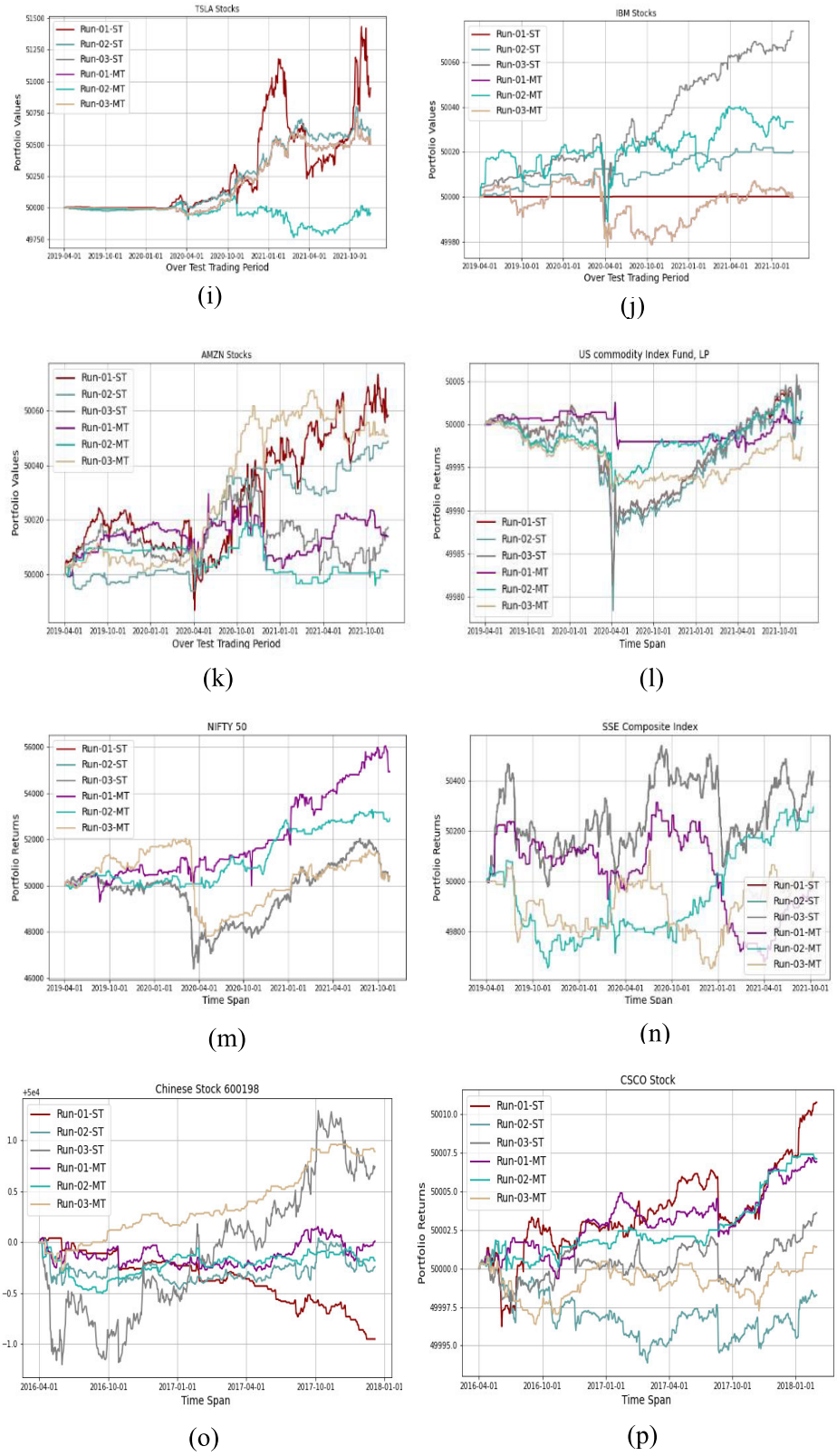


FIGURE 9. ***(Continued.)*** **Performance of RL trading agent in terms of Portfolio values over ten different stock markets (a,b,c,d,e,f,g,h) Portfolio values on TSLA, IBM, AMZN, SSE Composite Index, US Commodity Index, NIFTY 50 Index trends Chinese and CSCO with both single and multi-step ahead features and window size of 10 (,i,j,k,l,m,n,o,p) Portfolio values TSLA, IBM, AMZN, SSE Composite Index, US Commodity Index, NIFTY 50 Index, Chinese and CSCO future trends with both single and multi-step ahead features and window size of 15.**

TABLE 5. Results of RL agent in terms of profits and portfolio values over single Time Steps with window size 15.

Stock Name	Portfolio value	Total profit over trading profit[1]	Average Profit over trading period[1]	Loopback	Time step
Tesla	947.13	7043.28	1.555	1	Single
Tesla	620.94	4580.31	6.098	2	Single
Tesla	2333.81	11703.30	15.583	3	Single
Average	1300.62	7775.63	7.745	-	Single
IBM	0.11	49.78	0.066	1	Single
IBM	20.41	791.85	1.055	2	Single
IBM	73.65	815.37	1.085	3	Single
Average	31.39	552.33	0.735	-	Single
Amazon	58.13	882.78	1.175	1	Single
Amazon	48.70	799.03	1.06	2	Single
Amazon	17.15	886.19	1.18	3	Single
Average	41.32	856.0	1.133	-	Single
SSE Composite Index	434.00	9269.80	13.111	1	Single
SSE Composite Index	434.31	9270	13.3	2	Single
SSE Composite Index	434.31	9270	13.2	3	Single
Average	434.20	9269.933	13.20	-	Single
US commodity Index Fund, LP	3.99	94.5	0.13	1	Single
US commodity Index Fund, LP	18.42	129.02	0.18	2	Single
US commodity Index Fund, LP	-6.64	203	-0.27	3	Single
Average	5.2566	142.1733	0.0133	-	Single
NIFTY 50	283.74	40095.89	54.9258	1	Single
NIFTY 50	384.64	39911.2	54.9742	2	Single
NIFTY 50	384.64	54.9742	39911.29	3	Single
Average	351.006	26687.35	13340.39	-	Single
601988 (Stock-2008-2018)	-0.95	3.15	0.006	1	Single
601988 (Stock-2008-2018)	-0.24	5.64	0.01	2	Single
601988 (Stock-2008-2018)	0.74	12.59	0.02	3	Single
Average	-0.15	7.12667	0.012	-	Single
601988 (Stock-2005-2017)	11.95	814.49	0.38	1	Single
601988 (Stock-2005-2017)	-30.50	407.28	0.19	2	Single
601988 (Stock-2005-2017)	5.19	476.65	0.22	3	Single
Average	-4.4533	566.14	0.26333	-	Single
CSCO(Stock-2008-2018)	10.75	79.80	0.14	1	Single
CSCO(Stock-2008-2018)	-1.73	67.929	0.127	2	Single
CSCO(Stock-2008-2018)	3.59	63.34	0.111	3	Single
Average	4.2033	70.356	0.126	-	Single
CSCO(Stock-2000-2010)	-3.77	118.67	0.2358	1	Single
CSCO(Stock-2000-2010)	-2.70	122.94	0.244	2	Single
CSCO(Stock-2000-2010)	2.42	139.75	0.277	3	Single
Average	-1.35	127.12	0.25227	-	Single

[1] *This profit does not include loss during trading at different trading positions*

Commodity Index fund, and NIFTY50 with a window size of the previous time step set to 15 as an observing state. It is observed from the graphs that the loss of DQN reduces over different time steps of training. Moreover, the loss values are depicted in different colors for each experiment performed with different trials over both single and multi-step ahead features. More precisely, the curves are drawn with different colors indicating the different trials for each stock namely Tesla, IBM, Amazon, SSE composite index, U.S commodity index fund, NIFTY 50, Chinese Stock 60198, and CSCO stock.

D. DISCUSSIONS AND COMPARISONS

In this study, we proposed a deep reinforcement learning-based trading agent. The primary objective of the agent is set in such a way that it makes trading decisions over the trading period. The trading decisions should be performed very intelligently by the model that will ultimately increase profit at the end of the trading period. In the existing studies, different deep reinforcement learning-based agents are designed that observe the trading environment or the market situation before making trading decisions. These observations/states of the environment include different things such as historical prices, technical indicators of the historical data, the number of shares held, current portfolio balance, etc. In addition, some studies also involve the sentiments of the day, as well as closing and opening prices of the stock market as an observing state of the RL agent. It is evident from these research studies that the state engineering module of the RL agent is very important while making decisions E-g a human trader should know about the current situation of the stock market in different aspects so that an effective decision is

TABLE 6. Results of RL agent in terms of profits and portfolio values over single and multi time steps with window szie 15.

Stock Name	Portfolio value	Total profit over trading profit[1]	Average Profit over trading period[1]	Loopback	Time step
Tesla	-16.63	473.33	0.63	1	Multi
Tesla	502.58	4452.03	5.96	2	Multi
Tesla	-43.46	4813.14	6.45	3	Multi
Average	147.49	3246.16	4.346	-	Multi
IBM	-9.29	877.74	1.17	1	Multi
IBM	45.99	676.5	0.90	2	Multi
IBM	58.61	749.60	1.0048	3	Multi
Average	31.77	767.47	1.024	-	Multi
Amazon	13.94	596.24	0.7999	1	Multi
Amazon	50.60	822.78	1.1029	2	Multi
Amazon	1.08	447.34	0.600	3	Multi
Average	21.87	622.12	0.8344	-	Multi
SSE Composite Index	35.38	8802.168	12.450	1	Multi
SSE Composite Index	293.95	8193.698	11.5893	2	Multi
SSE Composite Index	36.69	8841.8	12.50	3	Multi
Average	122.006	8612.555	12.179	-	Multi
US commodity Index Fund, LP	0.17	99.159	0.136	1	Multi
US commodity Index Fund, LP	2.75	85.5499	0.1176	2	Multi
US commodity Index Fund, LP	6.47	103.6899	0.1426	3	Multi
Average	3.13	96.132	0.1320	-	Multi
NIFTY 50	408.65	33792.569	46.48	1	Multi
NIFTY 50	2890.56	35509.7	48.844	2	Multi
NIFTY 50	4925.79	64111.192	88.18	3	Multi
Average	2741.66	44470.85	61.168	-	Multi
600198 (Stock-2008-2018)	0.01	6.27	0.01	1	Multi
600198 (Stock-2008-2018)	0.11	6.92	0.01	2	Multi
600198 (Stock-2008-2018)	0.89	9.48	0.01	3	Multi
Average	0.33667	7.55667	0.01	-	Multi
600198 (Stock-2005-2017)	-11.19	475.80	0.223	1	Multi
600198 (Stock-2005-2017)	-3.17	415.19	0.19	2	Multi
600198 (Stock-2005-2017)	-13.76	529	0.24	3	Multi
Average	-9.37	473.33	0.217	-	Multi
CSCO(Stock-2008-2018)	6.90	79.78	0.14	1	Multi
CSCO(Stock-2008-2018)	7.10	88.56	0.166	2	Multi
CSCO(Stock-2008-2018)	1.40	78.56	0.144	3	Multi
Average	5.1333	82.3	0.15	-	Multi
CSCO(Stock-2000-2010)	-3.23	66.51	0.132	1	Multi
CSCO(Stock-2000-2010)	-8.03	112.13	0.22	2	Multi
CSCO(Stock-2000-2010)	2.08	139.6	0.277	3	Multi
Average	-3.06	106.08	0.209	-	Multi

[1] *This profit does not include loss during trading at different trading positions*

executed. In this study, the main hypothesis is that what if we make the trading agent that is aware of future prices?

The major reason behind this is that generally, stock prices are very fluctuating and very volatile in nature.

It is likely that future prices may fall, and if the decision to sell is made tomorrow, the corporation would make a loss. In addition, the future price trend is not already known in advance, hence for this reason we trained another model/forecasting network namely GRU that is able to forecast the future price of the stock, and that forecasted price is added to the RL agent's states to make it informed of the future when it takes action on that specific state. When the agent observes the current closing price, the forecasting network estimates the following day's price. This is one of the primary goals of the research, in which we develop a future-aware decision support system for algorithmic trading. Furthermore, the suggested RL agent's states include not only the future price but also certain other factors. More precisely, the states of the agent include the current portfolio balance, the number of holdings stocks it buys i-e inventory, current close price, and differences between current stock prices with previous historical prices, as well as the future

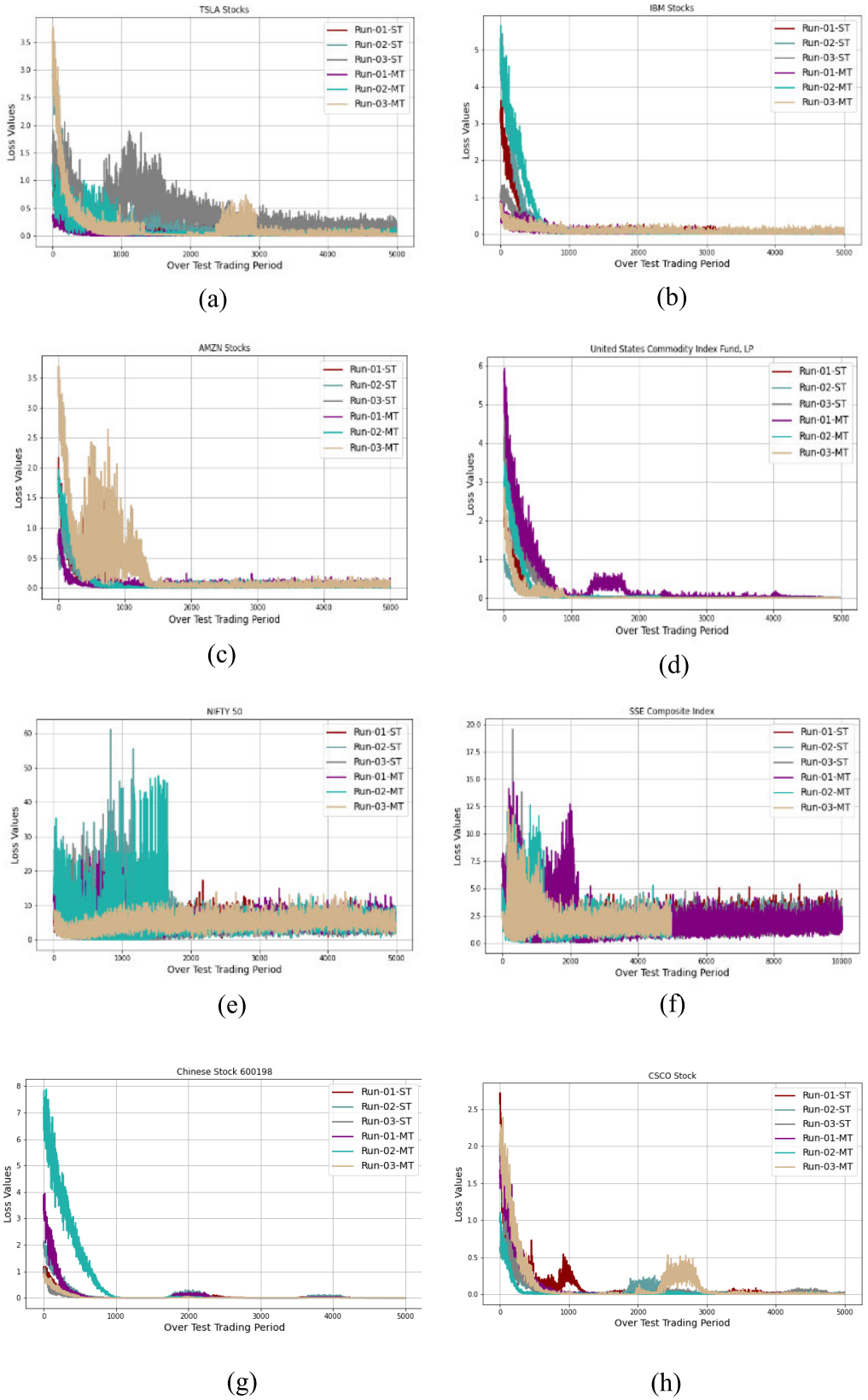


FIGURE 10. Performance of RL trading agent over three different stock markets (a,b,c,d,e,f,g,h) DQN Loss values on TSLA, IBM, AMZN, U.S commodity Index, SSE composite index, NIFTY 50 Index, Chinese and CSCO Stock market trends with both single and multi-step ahead features and window size of 10 (i,j,k,l,m,n,o,p) DQN loss values TSLA, IBM, AMZN, U.S commodity Index, SSE composite index, NIFTY 50 Index, Chinese and CSCO Stock market future trends with both single and multi-step ahead features and window size of 15.

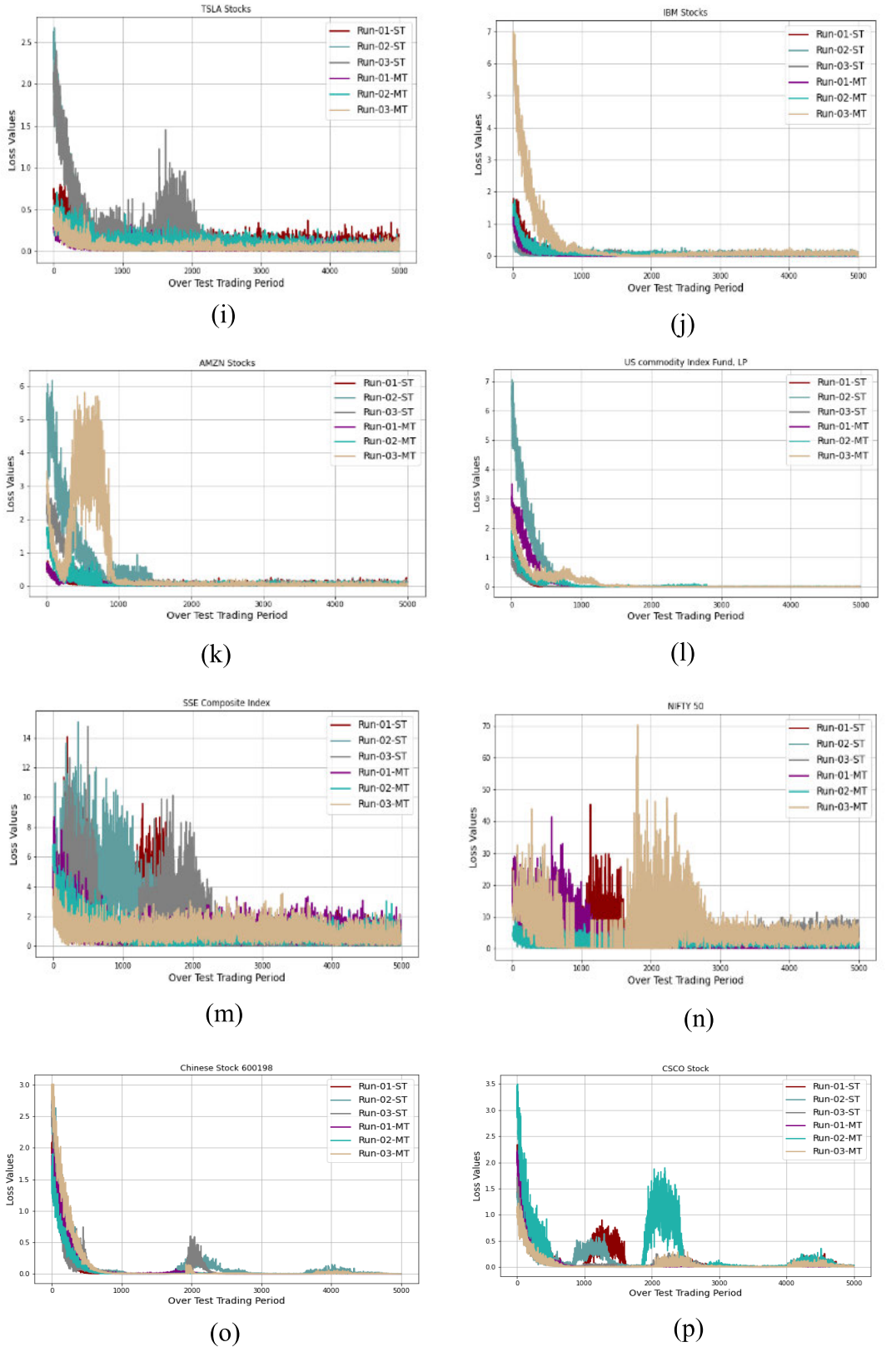


FIGURE 10. *(Continued.)* **Performance of RL trading agent over three different stock markets (a,b,c,d,e,f,g,h) DQN Loss values on TSLA, IBM, AMZN, U.S commodity Index, SSE composite index, NIFTY 50 Index, Chinese and CSCO Stock market trends with both single and multi-step ahead features and window size of 10 (i,j,k,l,m,n,o,p) DQN loss values TSLA, IBM, AMZN, U.S commodity Index, SSE composite index, NIFTY 50 Index, Chinese and CSCO Stock market future trends with both single and multi-step ahead features and window size of 15.**

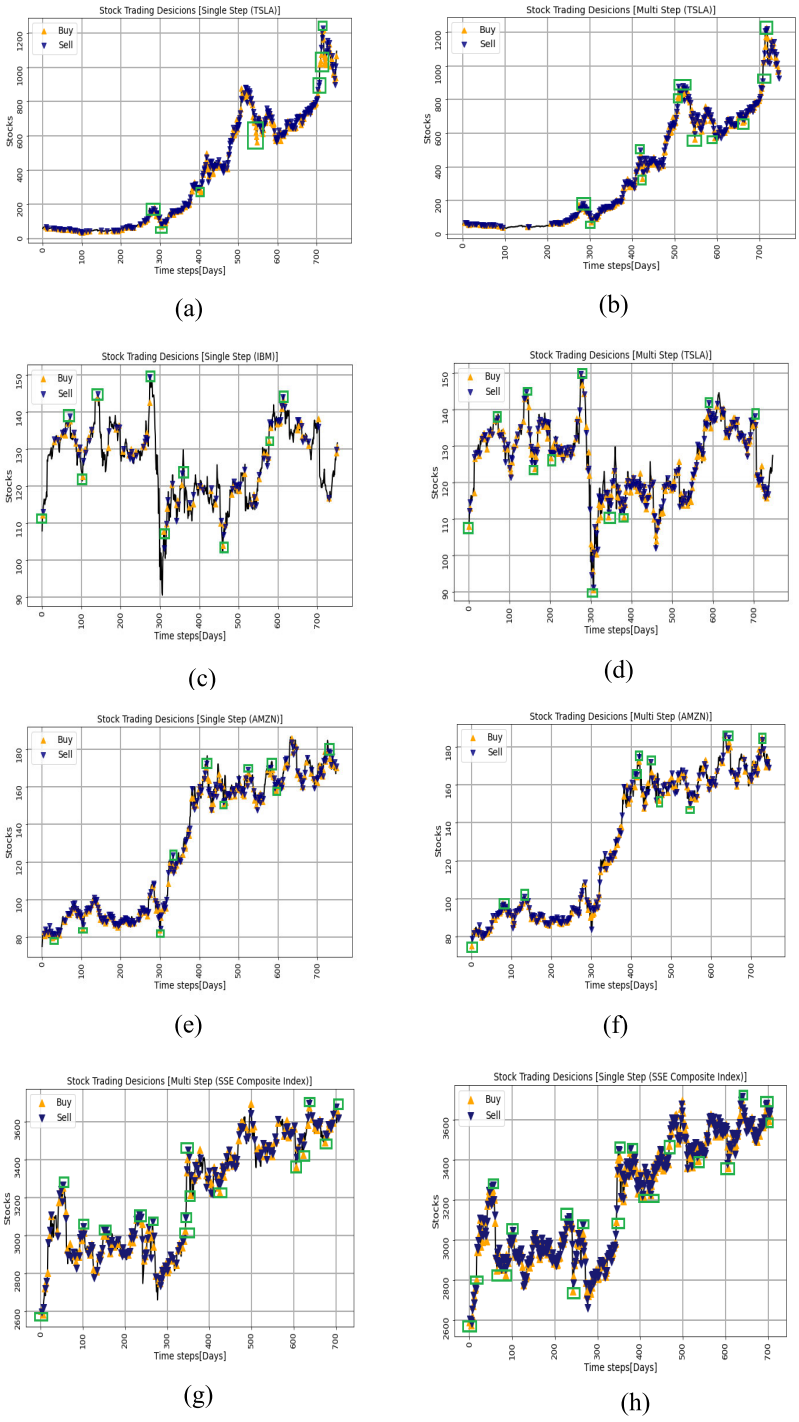


FIGURE 11. Performance of RL trading agent in terms of trading decisions over six different stock markets (a,b) Decisions on TSLA Stock market trends with both single and multi-step ahead features and window size of 15(c,d) Decisions on IBM Stock market future trends with both single and multi-step ahead features and window size of 15 (e,f) Decisions on Amazon Stock market future trend with both single and multi-step ahead features and window size of 15 (g,h) Decisions on SSE Composite Index future trend with both single and multi-step ahead features and window size of 15 (i,j) Decisions on US Commodity Index Fund future trend with both single and multi-step ahead features and window size of 15 (k,l) Decisions on NIFTY 50 Index future trend with both single and multi-step ahead features and window size of 15 (m,n) Decisions on Chinese 601988 stock (2018-2018) future trend with both single and multi-step ahead features and window size of 15 (o,p) Decisions on Chinese 601988 stock(2015-2017) future trend with both single and multi-step ahead features and window size of 15 (q,r) Decisions on CSCO stock (2018-2018) future trend with both single and multi-step ahead features and window size of 15 (s,t) Decisions on CSCO stock(2000-2010) future trend with both single and multi-step ahead features and window size of 15.

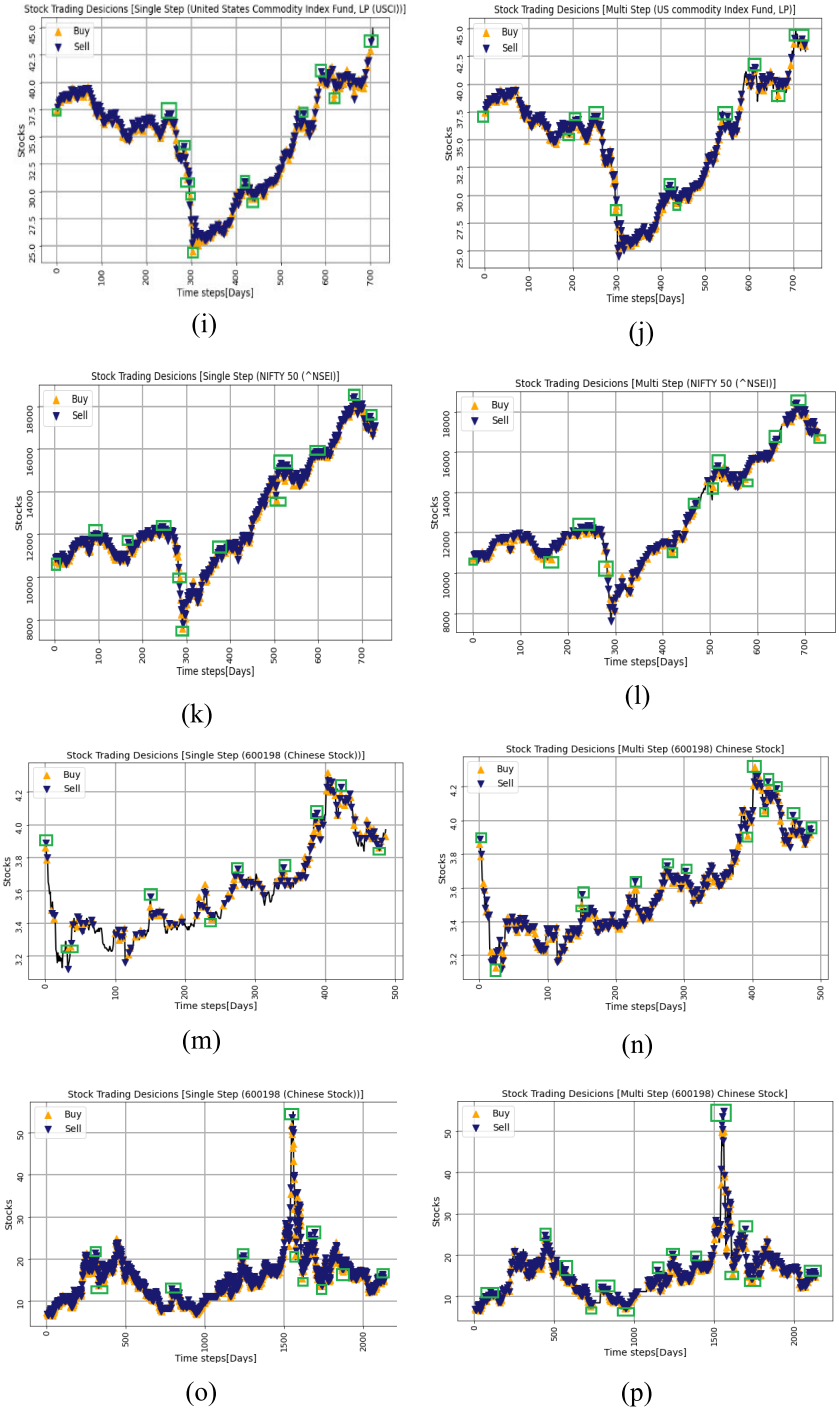


FIGURE 11. ***(Continued.)*** **Performance of RL trading agent in terms of trading decisions over six different stock markets (a,b) Decisions on TSLA Stock market trends with both single and multi-step ahead features and window size of 15(c,d) Decisions on IBM Stock market future trends with both single and multi-step ahead features and window size of 15 (e,f) Decisions on Amazon Stock market future trend with both single and multi-step ahead features and window size of 15 (g,h) Decisions on SSE Composite Index future trend with both single and multi-step ahead features and window size of 15 (i,j) Decisions on US Commodity Index Fund future trend with both single and multi-step ahead features and window size of 15 (k,l) Decisions on NIFTY 50 Index future trend with both single and multi-step ahead features and window size of 15 (m,n) Decisions on Chinese 601988 stock (2018-2018) future trend with both single and multi-step ahead features and window size of 15 (o,p) Decisions on Chinese 601988 stock(2015-2017) future trend with both single and multi-step ahead features and window size of 15 (q,r) Decisions on CSCO stock (2018-2018) future trend with both single and multi-step ahead features and window size of 15 (s,t) Decisions on CSCO stock(2000-2010) future trend with both single and multi-step ahead features and window size of 15.**

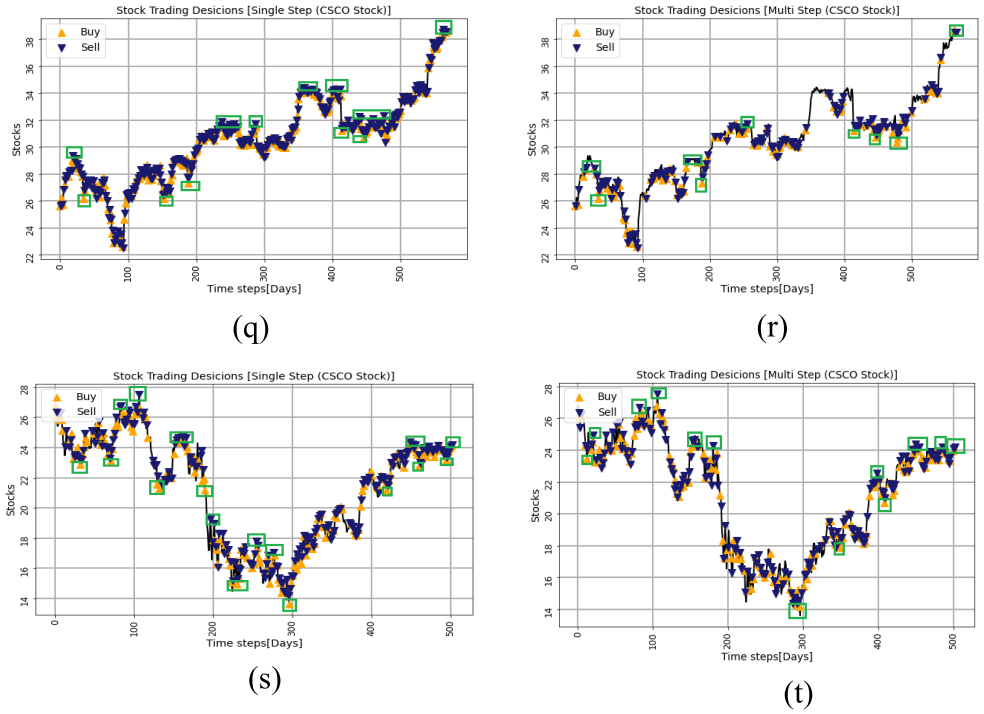


FIGURE 11. ***(Continued.)*** **Performance of RL trading agent in terms of trading decisions over six different stock markets (a,b) Decisions on TSLA Stock market trends with both single and multi-step ahead features and window size of 15(c,d) Decisions on IBM Stock market future trends with both single and multi-step ahead features and window size of 15 (e,f) Decisions on Amazon Stock market future trend with both single and multi-step ahead features and window size of 15 (g,h) Decisions on SSE Composite Index future trend with both single and multi-step ahead features and window size of 15 (i,j) Decisions on US Commodity Index Fund future trend with both single and multi-step ahead features and window size of 15 (k,l) Decisions on NIFTY 50 Index future trend with both single and multi-step ahead features and window size of 15 (m,n) Decisions on Chinese 601988 stock (2018-2018) future trend with both single and multi-step ahead features and window size of 15 (o,p) Decisions on Chinese 601988 stock(2015-2017) future trend with both single and multi-step ahead features and window size of 15 (q,r) Decisions on CSCO stock (2018-2018) future trend with both single and multi-step ahead features and window size of 15 (s,t) Decisions on CSCO stock(2000-2010) future trend with both single and multi-step ahead features and window size of 15.**

price. We do many experiments using a single step ahead of future prices or the next two following future prices, i.e., multi-step. These states are observed by the agent to take the decision to buy, sell and hold the stocks. These decisions are viewed as actions of the agent, which is awarded a positive reward if the stock is sold for a profit and 0 rewards as a penalty if it makes a loss. The overall objective of the agent is to maximize the profit while making decisions.

Moreover, if we analyze the results of the GRU model i-e forecasting network shows very good performance in terms of RMSE, MSE, and MAE errors. It signifies that the GRU model accurately predicts future prices, and that the agent is fully aware of the future. Since it is critical to more accurately foresee the future, if the GRU forecast is incorrect, it will also mislead the RL agent. In addition, the proposed GRU is comprised of less trainable parameters having only 4 units and 1 dense layer. This will make it more viable for use in real-time applications when time is of the essence. In general, lightweight applications or models are easier to deploy in practical scenarios. To add more, if we analyze the results of the RL agent for stock trading decisions then it is also very encouraging. As seen in Figures 8 and 11, where the trading decisions are illustrated, the model buys when the stock values are at lower peaks and sells. the stock values are at their greatest peaks to gain a profit. As the trading period progresses, the portfolio values grow, as seen in Figure 9. Furthermore, the RL agent has difficulty with IBM stocks since their prices fluctuate significantly.

The DQN network, which is an agent based on a neural network, is optimized and converges fast, as shown by the loss graphs in Figure 10. Moreover, there exists a very deep relationship between stock market returns and risk. Minimizing the risk while maximizing the returns is one of the most difficult challenges for profitable trading. Each investment strategy entails some degree of risk for the investor during trading. Minimizing the risk while trading, as well as increasing returns, is one of the good research areas and problems in the risk management field. Hence, the comparison between benchmark methods in terms of risk-based measures including Buy and Hold, Moving Average, and Signal rolling is also made as shown in Table 7. These measures indicate the risk-adjusted returns. More precisely, when comparing

TABLE 7. Comparison with existing methods in terms of risk-based measures.

Method	Dataset	Sharpe ratio	Sortino ratio
Proposed	**601988 stocks of Chinese**	**0.305**	**0.417**
Proposed	**CSCO Stock**	**0.564**	**0.662**
Proposed	**IBM**	**0.699**	**0.502**
Proposed	**TSLA**	1.011	**3.691**
Proposed	**Amazon**	**0.673**	0.644
Proposed	**SSE**	**0.884**	0.734
Proposed	**Nifty50 Index**	0.823	0.505
Proposed	**US Commodity**	**0.559**	**0.607**
Moving Average	601988 Chinese stock	-0.097	-0.046
Moving Average	CSCO Stock	0.096	0.042
Moving Average	IBM	0.067	0.030
Moving Average	TSLA	-0.053	-0.018
Moving Average	Amazon	-0.269	-0.090
Moving Average	SSE Composite Index	-0.033	-0.157
Moving Average	NIFTY 50 Index	0.472	0.839
Moving Average	U.S commodity Index	-0.176	-0.072
Buy and Hold	601988 Chinese stock	0.01	0.01
Buy and Hold	CSCO Stock	0.01	0.02
Buy and Hold	IBM	0.21	0.28
Buy and Hold	TSLA	**1.68**	2.53
Buy and Hold	Amazon	0.92	**1.35**
Buy and Hold	SSE Composite Index	0.01	0.02
Buy and Hold	NIFTY 50 Index	0.68	0.88
Buy and Hold	U.S commodity Index	0.45	0.57
Signal rolling	601988 Chinese stock	-0.097	-0.052
Signal rolling	CSCO Stock	-0.062	-0.035
Signal rolling	IBM	-0.035	-0.020
Signal rolling	TSLA	0.069	0.035
Signal rolling	Amazon	-0.025	-0.013
Signal rolling	SSE Composite Index	0.267	1.809
Signal rolling	NIFTY 50 Index	**0.851**	**1.742**
Signal rolling	U.S commodity Index	-0.055	-0.033

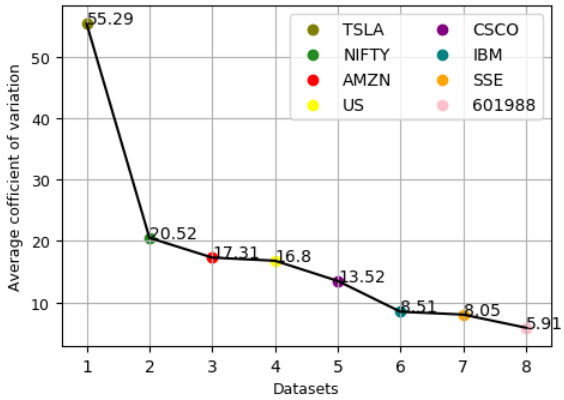


FIGURE 12. Average coefficient of variations of all datasets.

the performance of trading strategies, the rate of return is not the only measure to indicate the performance of the underlying method. However, measuring return rates along with the degree of risk is of greater importance.

This concept is defined as computing the return rates by involving the information that how much risk is associated with obtaining that return. Here, we have considered the Sharpe ratio and Sortino ratio as risk-based measures. It is observed from the Table 7 we have achieved good values of risk-based measures in comparison with standard methods. However, on some datasets, the signal rolling and Buy and Hold methods perform well. The average coefficient of variation of all datasets is also given in Figure 12. In addition, in this method, the RL agent receives the reward in terms of profits achieved over the trading periods, however, rewards are not on the basis that how much the RL agent can deal with risk management while taking action of sell, buy and hold. This might also be a limitation of the proposed RL and can be further improved in the future in which rewards are designed on the basis of risk-adjusted returns. As a result of adding this, the RL agent's performance in terms of risk-based metrics is improved much more.

Further, it is logically concluded that automated trading-based decision support systems that use a computer plan to

TABLE 8. Comparison of Proposed and Existing Studies.

Existing Studies	RL Agent	States	Future Price	Stocks Data	Results
Li *et al.*[31]	Double and Dueling DQN	Stock data features, technical indicators and candlestick charts	No	S&P500 Stock and Chinese Stock market	PR= (162.87-Chinese Stock-)PR=(212.96 S&P500)
Wu *et al.*[66]	LSTM based DRL Agent	Volume, Raw Historical prices and Technical indicators	No	Stocks of Chinese Stock markets	Highest average profit=(1, 820, 798 CNY)
Xing *et al.* [67]	GDQN and GDPG	Historical prices and Technical indicators of stock markets	No	U.S, U.K and Chinese Stock markets	Highest R (%)=46.0%
Nan *et al.* [52]	DQN	Cash, Holding stocks, Current open price, Average differences of today and open prices with previous 5/50 days, and news headlines sentiments	No	MSFT, Amazon and Tesla Stocks	Highest Sharpe ratio= (2.432)
Yuming *et al.*[32]	DQN, Dueling, and Double DQN	Historical closing price	No	U.S Stock markets	Highest Profit= (1029)
Li *et al.*[70]	Suggested DQN and A3C extended version	Market variables, Technical indicators, trading cash and previous Sharpe ratio	No	APPL, IBM and PG from NASDAQ	Highest AR= (85.33%)
Carta *et al.*[71]	Double Deep Q Networks	Image based market stock prices	No	Standard & Poor's 500 and J.P. Morgan and Microsoft stocks	Highest AR= (23.20%), Return=(1265.50)
Akhil *et al.*[33]	DDPG	Historical Stock prices, and news headline sentiments	No	NASDAQ-GE and NASDAQ-GOOG	-
Souradeep Chakraborty[72]	DQN	Market features and technical indicators	No	Forex market	Sharpe ratio= (1.79)
Proposed	**DQN**	**Current portfolio balance, Number of Holding stocks, Current price and its difference with previous n days price, Single and multi-step ahead forecasted price**	**Yes**	**IBM, Amazon, Tesla SSE Composite Index, U.S commodity Index, NIFTY 50 Index, Chinese Stock, CSCO stock**	**Highest Profit=99580.42 on NIFTY 50 Index) Highest PR=4925.19**

develop buy and sell signals and execute transactions based on a specified trading plan or trading rationale are very crucial to assist investors. The proposed algorithmic trading system referred to as the decision support system offers numerous advantages over human traders, including a more reliable system, speed and accuracy execution, and the ability to be unaffected by emotional factors. There exist limitations to conventional human trading strategies. For instance, the information cannot be precisely valued, and the summarized markers and fixed operating models cannot adaptively cope with alterations in the environments. In such cases, automated trading systems that are less likely to be influenced by external factors are more advantageous to investors and policymakers when making decisions. Different such systems are previously designed to support the finance field, however, in this work, we add to the literature by introducing another notion for enhancing automated trading systems by incorporating future patterns. This research also encourages researchers and scientists to consider in terms of designing future-aware systems. Although it is not always feasible to predict the future accurately, but the possibility exists. As a result, companies suffer less from losses and risks. More specifically, a point-by-point comparison is added between this work and existing studies as shown in Table 8.

In Table 8, we have mentioned the RL agents or algorithms that are employed in the previous studies as well as states, future prices, and data of stocks. It is clear that deep-Q-Network is used for algorithmic trading decisions in the majority of studies; nevertheless, some researchers have used sophisticated and extended versions such as Dueling DQN and Double DQN. Furthermore, differences are also noticed in the construction of states, such as one fascinating research by Li et al. [31] in which candlestick charts are used as an image feature. Similarly, Nan et al. [52] incorporate the news headline's sentiments about the state in addition to cash, holding stocks, current open price, and

average differences between today and open prices with previous 5/50 days. Azhikodan et al. [33] also employs the new headlines sentiments nevertheless, instead of the DQN model, they employed a DDPG-based RL agent. Following that, if we look at stock data, we observe that several research studies have conducted experiments on various stock markets, most notably the US and Chinese stock markets, as well as IBM, TSLA, MSFT, and AMZN-type popular markets. Subsequently, if we observe the results in terms of evaluation metrics, then different research studies evaluate the proposed methods with different criteria such as profits, annualized cumulative return, Sharpe ratio, average profits, Sortino ratios, etc. In Table 8, we have listed the highest results in all of their experimentations. Table 8 summarizes the existing research in light of this study's contributions. More specifically, this research proposed another viewpoint or perspective in which future prices are viewed as state information. As observed in Table 8, this is not being studied in existing studies. In addition, this study performed the experiment on different markets that are also being used in existing studies such as TSLA, Amazon, IBM, Chinese stock and CSCO. Moreover, the RL-agent is also tested on equity and commodity markets i.e., SSE Composite Index, NIFTY50, and US Commodity.

In addition to the above discussion, it is essential to acknowledge the limitations to initiate further research. One possible limitation of the study is that RL agents employ only the close prices, however, what if further information is added to the state i.e., instead of using noisy closing prices, technical indicators based on stock prices trends can be employed, or what if the sentiments of the day or any other information's such as a number of shares hold, open and close prices are incorporated to make the agent more informed about the market situation. Secondly, some more complex rewards function can also be designed e-g by only focusing on profit, the Sharpe ratio, and Sortino ratio that involves risk factors should also need to be modeled as reward functions along with future trend module to further enhance the systems.

V. CONCLUSION

In automated trading frameworks, building lucrative trading technique is very crucial in which the computer program or algorithm monitors and implements the trading decisions regarding stocks. A significant research challenge in financial market trading is to develop automated trading systems that make profitable decisions. In this paper, one such decision support system for automated stock market trading is proposed to perform optimal decisions making regarding stock selling and buying. The proposed model integrates both deep learning and reinforcement learning in which both past historical trends of stocks, as well as future trends, are amalgamated during decisions. A forecasting network model is proposed that estimates the future situations of stocks concurrently which is then combined with past histories. In addition, the forecasting network is based on GRU to capture a more vital aspect of time series data. The model is evaluated on different stock market data and shows the good profit values which is one of the primary goals of the agent. The suggested decision support system will aid investors, policymakers, and all other business operations by advising them whether to purchase, hold, or sell stocks. Different other technical and fundamental indicators, both past and future, will be used in future research to improve the performance of the model.

REFERENCES

[1] I. Diakoulakis, D. Koulouriotis, and D. Emiris, "A review of stock market prediction using computational methods," *Comput. Methods Decision-Making, Econ. Finance*, vol. 74, pp. 379–403, Aug. 2002.

[2] S. Lahmiri and S. Bekiros, "The impact of COVID-19 pandemic upon stability and sequential irregularity of equity and cryptocurrency markets," *Chaos, Solitons Fractals*, vol. 138, Sep. 2020, Art. no. 109936.

[3] J. Markard and D. Rosenbloom, "A tale of two crises: COVID-19 and climate," *Sustainability: Sci., Pract. Policy*, vol. 16, no. 1, pp. 53–60, Dec. 2020.

[4] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," *J. Finance Data Sci.*, vol. 2, no. 1, pp. 42–57, 2016.

[5] M. Bukhari, S. Yasmin, S. Sammad, and A. A. A. El-Latif, "A deep learning framework for leukemia cancer detection in microscopic blood samples using squeeze and excitation learning," *Math. Problems Eng.*, vol. 2022, pp. 1–18, Jan. 2022.

[6] R. Ashraf, S. Afzal, A. U. Rehman, S. Gul, J. Baber, M. Bakhtyar, I. Mehmood, O.-Y. Song, and M. Maqsood, "Region-of-Interest based transfer learning assisted framework for skin cancer detection," *IEEE Access*, vol. 8, pp. 147858–147871, 2020.

[7] M. Bukhari, K. B. Bajwa, S. Gillani, M. Maqsood, M. Y. Durrani, I. Mehmood, H. Ugail, and S. Rho, "An efficient gait recognition method for known and unknown covariate conditions," *IEEE Access*, vol. 9, pp. 6465–6477, 2021.

[8] M. Sharma, C. J. Kumar, and A. Deka, "Early diagnosis of Rice plant disease using machine learning techniques," *Arch. Phytopathol. Plant Protection*, vol. 55, no. 3, pp. 259–283, Feb. 2022.

[9] T. T. Nguyen, T. D. Hoang, M. T. Pham, T. T. Vu, T. H. Nguyen, Q.-T. Huynh, and J. Jo, "Monitoring agriculture areas with satellite images and deep learning," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106565.

[10] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for Youtube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.

[11] S. Yasmin, M. Y. Durrani, S. Gillani, M. Bukhari, M. Maqsood, and M. Zghaibeh, "Small obstacles detection on roads scenes using semantic segmentation for the safe navigation of autonomous vehicles," *J. Electron. Imag.*, vol. 31, no. 6, Apr. 2022, Art. no. 061806.

[12] D. Kumar, P. K. Sarangi, and R. Verma, "A systematic review of stock market prediction using machine learning and statistical techniques," *Mater. Today, Proc.*, vol. 49, pp. 3187–3191, Jan. 2022.

[13] I. Parmar, "Stock market prediction using machine learning," in *Proc. 1st Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, 2018, pp. 574–576.

[14] H. Chung and K.-S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability*, vol. 10, no. 10, p. 3765, Oct. 2018.

[15] S. I. Lee and S. J. Yoo, "Multimodal deep learning for finance: Integrating and forecasting international stock markets," *J. Supercomput.*, vol. 76, no. 10, pp. 8294–8312, Oct. 2020.

[16] V. Ingle and S. Deshmukh, "Ensemble deep learning framework for stock market data prediction (EDLF-DP)," *Global Transitions Proc.*, vol. 2, no. 1, pp. 47–66, Jun. 2021.

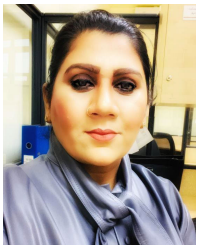
[17] J. B. Chakole, M. S. Kolhe, G. D. Mahapurush, A. Yadav, and M. P. Kurhekar, "A Q-learning agent for automated trading in equity stock markets," *Expert Syst. Appl.*, vol. 163, Jan. 2021, Art. no. 113761.

[18] R. C. J. Chia, S. Y. Lim, P. K. Ong, and S. F. Teh, "Pre and post Chinese new year holiday effects: Evidence from Hong Kong stock market," *Singap. Econ. Rev.*, vol. 60, no. 4, Sep. 2015, Art. no. 1550023.

[19] Q. Huang, T. Wang, D. Tao, and X. Li, "Biclustering learning of trading rules," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2287–2298, Oct. 2015.

[20] P. Treleaven, M. Galas, and V. Lalchand, "Algorithmic trading review," *Commun. ACM*, vol. 56, pp. 76–85, Nov. 2013.
[21] Y. Yadav, "How algorithmic trading undermines efficiency in capital markets," *Vanderbilt Law Rev.*, vol. 68, no. 6, p. 1607, Jan. 2015.
[22] D. Andriosopoulos, M. Doumpos, P. M. Pardalos, and C. Zopounidis, "Computational approaches and data analytics in financial services," *J. Oper. Res. Soc.*, vol. 70, no. 10, pp. 1579–1580, Oct. 2019.
[23] T. L. Meng and M. Khushi, "Reinforcement learning in financial markets," *Data*, vol. 4, no. 3, p. 110, 2019.
[24] M. H. Miller, J. Muthuswamy, and R. E. Whaley, "Mean reversion of standard & Poor's 500 index basis changes: Arbitrage-induced or statistical illusion?" *J. Finance*, vol. 49, pp. 479–513, Jun. 1994.
[25] J.-L. Wang and S.-H. Chan, "Stock market trading rule discovery using pattern recognition and technical analysis," *Expert Syst. Appl.*, vol. 33, no. 2, pp. 304–315, Aug. 2007.
[26] D. Jothimani and S. S. Yadav, "Stock trading decisions using ensemble-based forecasting models: A study of the Indian stock market," *J. Banking Financial Technol.*, vol. 3, no. 2, pp. 113–129, Oct. 2019.
[27] Y. Hu, B. Feng, X. Zhang, E. W. T. Ngai, and M. Liu, "Stock trading rule discovery with an evolutionary trend following model," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 212–222, Jan. 2015.
[28] N. Metawa, M. K. Hassan, S. Metawa, and M. F. Safa, "Impact of behavioral factors on investors' financial decisions: Case of the Egyptian stock market," *Int. J. Islamic Middle Eastern Finance Manage.*, vol. 12, no. 1, pp. 30–55, Mar. 2019.
[29] L. Zhou and J. Huang, "Investor trading behaviour and stock price crash risk," *Int. J. Finance Econ.*, vol. 24, no. 1, pp. 227–240, Jan. 2019.
[30] S. F. Shah, M. Alshurideh, B. A. Kurdi, and S. A. Salloum, "The impact of the behavioral factors on investment decision-making: A systemic review on financial institutions," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.*, 2020, pp. 100–112.
[31] Y. Li, P. Liu, and Z. Wang, "Stock trading strategies based on deep reinforcement learning," *Sci. Program.*, vol. 2022, pp. 1–15, Mar. 2022.
[32] Y. Li, P. Ni, and V. Chang, "Application of deep reinforcement learning in stock trading strategies and stock forecasting," *Computing*, vol. 102, no. 6, pp. 1305–1322, Jun. 2020.
[33] A. R. Azhikodan, A. G. Bhat, and M. V. Jadhav, "Stock trading bot using deep reinforcement learning," in *Innovations in Computer Science and Engineering*. Cham, Switzerland: Springer, 2019, pp. 41–49.
[34] M. Metghalchi, N. Durmaz, P. Cloninger, and K. Farahbod, "Trading rules and excess returns: Evidence from Turkey," *Int. J. Islamic Middle Eastern Finance Manage.*, vol. 14, no. 4, pp. 713–731, Jul. 2021.
[35] C. Tudor and A. Anghel, "The financialization of crude oil markets and its impact on market efficiency: Evidence from the predictive ability and performance of technical trading strategies," *Energies*, vol. 14, no. 15, p. 4485, Jul. 2021.
[36] M. Arif, M. Hasan, J. A. Tunio, A. B. Naeem, and M. Zia-Ullah, "Testing success of moving averages in Pakistan stock exchange: Post financial liberalization era in concern," *Abasyn J. Social Sci.*, pp. 426–433, 2017.
[37] S. Boonpeng and P. Jeatrakul, "Decision support system for investing in stock market by using OAA-neural network," in *Proc. 8th Int. Conf. Adv. Comput. Intell. (ICACI)*, Feb. 2016, pp. 1–6.
[38] C. Liu and H. Malik, "A new investment strategy based on data mining and neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 3094–3099.
[39] A. Rodríguez-González, Á. García-Crespo, R. Colomo-Palacios, F. Guldrís Iglesias, and J. M. Gómez-Berbís, "CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator," *Expert Syst. Appl.*, vol. 38, no. 9, pp. 11489–11500, Sep. 2011.
[40] A. K. M. A. Ullah, F. Imtiaz, M. U. M. Ihsan, M. G. R. Alam, and M. Majumdar, "Combining machine learning and effective feature selection for real-time stock trading in variable time-frames," 2021, *arXiv:2107.13148*.
[41] X. Zhang, Y. Hu, K. Xie, W. Zhang, L. Su, and M. Liu, "An evolutionary trend reversion model for stock trading rule discovery," *Knowl.-Based Syst.*, vol. 79, pp. 27–35, May 2015.
[42] M. Qiu, Y. Song, and F. Akagi, "Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market," *Chaos, Solitons Fractals*, vol. 85, pp. 1–7, Apr. 2016.
[43] X. Zhong and D. Enke, "Forecasting daily stock market return using dimensionality reduction," *Expert Syst. Appl.*, vol. 67, pp. 126–139, Jan. 2017.
[44] J. E. Moody, M. Saffell, Y. Liao, and L. Wu, "Reinforcement learning for trading systems and portfolios," in *Proc. KDD*, 1998, pp. 279–283.
[45] J. Carapuço, R. Neves, and N. Horta, "Reinforcement learning applied to Forex trading," *Appl. Soft Comput.*, vol. 73, pp. 783–794, Dec. 2018.
[46] J. Sun, H. Fujita, P. Chen, and H. Li, "Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble," *Knowl.-Based Syst.*, vol. 120, pp. 4–14, Mar. 2017.
[47] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, "Class-imbalanced dynamic financial distress prediction based on AdaBoost-SVM ensemble combined with SMOTE and time weighting," *Inf. Fusion*, vol. 54, pp. 128–144, Feb. 2020.
[48] Y. Deng, Y. Kong, F. Bao, and Q. Dai, "Sparse coding-inspired optimal trading system for HFT industry," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 467–475, Apr. 2015.
[49] L. Troiano, E. M. Villa, and V. Loia, "Replicating a trading strategy by means of LSTM for financial industry applications," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3226–3234, Jul. 2018.
[50] H. Yang, X.-Y. Liu, S. Zhong, and A. Walid, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in *Proc. 1st ACM Int. Conf. AI Finance*, 2020, pp. 1–8.
[51] X.-Y. Liu, Z. Xiong, S. Zhong, H. Yang, and A. Walid, "Practical deep reinforcement learning approach for stock trading," 2018, *arXiv:1811.07522*.
[52] A. Nan, A. Perumal, and O. R. Zaiane, "Sentiment and knowledge based algorithmic trading with deep reinforcement learning," 2020, *arXiv:2001.09403*.
[53] M.-Y. Chen, "A high-order fuzzy time series forecasting model for internet stock trading," *Future Gener. Comput. Syst.*, vol. 37, pp. 461–467, Jul. 2014.
[54] S. Lauguico, R. C. Ii, J. Alejandrino, D. Macasaet, R. R. Tobias, A. Bandala, and E. Dadios, "A fuzzy logic-based stock market trading algorithm using bollinger bands," in *Proc. IEEE 11th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ., Manage. (HNICEM)*, Nov. 2019, pp. 1–6.
[55] A. Rakićević, V. Simeunović, B. Petrović, and S. Milić, "An automated system for stock market trading based on logical clustering," *Tehnički vjesnik*, vol. 25, no. 4, pp. 970–978, 2018.
[56] Y. Kim, W. Ahn, K. J. Oh, and D. Enke, "An intelligent hybrid trading system for discovering trading rules for the futures market using rough sets and genetic algorithms," *Appl. Soft Comput.*, vol. 55, pp. 127–140, Jun. 2017.
[57] D. Silver, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[58] Z. Li, S. Xue, W. Lin, and M. Tong, "Training a robust reinforcement learning controller for the uncertain system based on policy gradient method," *Neurocomputing*, vol. 316, pp. 313–321, Nov. 2018.
[59] C. Li, X. Wei, Y. Zhao, and X. Geng, "An effective maximum entropy exploration approach for deceptive game in reinforcement learning," *Neurocomputing*, vol. 403, pp. 98–108, Aug. 2020.
[60] C. Ma, Z. Li, D. Lin, and J. Zhang, "Parallel multi-environment shaping algorithm for complex multi-step task," *Neurocomputing*, vol. 402, pp. 323–335, Aug. 2020.
[61] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, nos. 1–2, pp. 99–134, 1998.
[62] V. Mnih, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[63] I. Munemasa, Y. Tomomatsu, K. Hayashi, and T. Takagi, "Deep reinforcement learning for recommender systems," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Mar. 2018, pp. 226–233.
[64] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3389–3396.
[65] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.
[66] J. Wu, C. Wang, L. Xiong, and H. Sun, "Quantitative trading on stock market based on deep reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[67] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, and H. Fujita, "Adaptive stock trading strategies with deep reinforcement learning methods," *Inf. Sci.*, vol. 538, pp. 142–158, Oct. 2020.

[68] T. Théate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114632.
[69] H. Jia, X. Zhang, J. Xu, W. Zeng, H. Jiang, X. Yan, and J.-R. Wen, "Variance reduction for deep Q-learning using stochastic recursive gradient," 2020, *arXiv:2007.12817*.
[70] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108014–108022, 2019.
[71] S. Carta, A. Corriga, A. Ferreira, A. S. Podda, and D. R. Recupero, "A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning," *Int. J. Speech Technol.*, vol. 51, no. 2, pp. 889–905, Feb. 2021.
[72] S. Chakraborty, "Capturing financial markets to apply deep reinforcement learning," 2019, *arXiv:1907.04373*.



YASMEEN ANSARI is currently an Assistant Professor with the Department of Finance, College of Administrative and Financial Sciences, Riyadh, Saudi Arabia. Her teaching experience spans over 19 years. Her research interests include financial literacy, behavioral finance, e-learning, and research tools and techniques.

SADAF YASMIN received the B.S. degree in software engineering from the NUML (APCOMS), Islamabad, and the M.S. and Ph.D. degrees in computer science from the Capital University of Science and Technology, Islamabad. She is currently working as an Assistant Professor with the Department of Computer Science, COMSATS University Islamabad, Attock Campus, Pakistan. She worked on several research projects during and after her Ph.D. degree. Her research interests include network protocol design, computer vision, medical imaging, and pattern recognition. She is also serving as a reviewer for various reputed journals.

SHENEELA NAZ received the Ph.D. degree in computer science with specialization in multimedia and communication from the Capital University of Science and Technology (CUST), Islamabad, Pakistan, in 2018. She is currently working as an Assistant Professor with COMSATS University Islamabad, Islamabad. She has experience of more than six years in teaching, research, and industry research and development. She is the author of several publications in internationally recognized peer-reviewed journals and conferences. Her research interests include information-centric networks, network protocols and architectures, data science, and cloud computing.

HIRA ZAFFAR received the M.S. degree in computer science with a keen interest in machine learning and data sciences. She is currently working as a Lecturer with the Department of Computer Science, Air University, Aerospace and Aviation Kamra Campus, Pakistan. Her research interests include latest machine learning algorithms to develop automated solutions, especially in the field of educational data mining, pattern recognition, and data analytics.

ZEESHAN ALI received the Ph.D. degree in computer science with a keen interest in the application of artificial intelligence to develop enterprise-scale mission-critical systems. He is currently working as the Director of the Research and Development Setups, National University of Computer and Emerging Sciences. He has hands-on experience in leading and delivering predictive modeling and data mining projects. His research interests include latest machine learning and deep learning algorithms to solve real-world problems. The areas of his interest are medical image analysis, recommender systems, stock exchange prediction, and big data analytics.

JIHOON MOON received the Ph.D. degree in electrical and computer engineering from Korea University, Seoul, South Korea, in 2021. From 2011 to 2013, he was a Social Service Agent with Seoul Metropolitan Rapid Transit (SMRT) Corporation (merged with Seoul Transportation Corporation, in 2017), Seoul. From June 2021 to August 2022, he was a Postdoctoral Researcher with Chung-Ang University, Seoul. He has been working as a member of the faculty with the Department of AI and Big Data, Soonchunhyang University, Asan, South Korea, since September 2022, and serving as a Topical Advisory Panel Member for sustainability. His research interests include time-series analysis, energy forecasting, and machine learning applications in different industries.

SEUNGMIN RHO is currently an Associate Professor with the Department of Industrial Security, Chung-Ang University. His current research interests include database, big data analysis, music retrieval, multimedia systems, machine learning, knowledge management, and computational intelligence. He has published 300 papers in refereed journals and conference proceedings in these areas. He has been involved in more than 20 conferences and workshops as various chairs and more than 30 conferences/workshops as a program committee member. He has edited a number of international journal special issues as a Guest Editor, such as *Multimedia Systems*, *Information Fusion*, and *Engineering Applications of Artificial Intelligence*.