2022 RIVF International Conference on Computing and Communication Technologies (RIVF) | 978-1-6654-6166-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/RIVF55975.2022.10013861

Deep Reinforcement Learning methods for Automation Forex Trading

Tan Chau[*,†], Minh-Tri Nguyen[*,†], Duc-Vu Ngo[*,†], Anh-Duc T. Nguyen[*,†],Trong-Hop Do[*,†]

[*] Faculty of Information Science and Engineering, University of Information Technology, Ho Chi Minh City, Vietnam.
[†] Vietnam National University, Ho Chi Minh City, Vietnam.

***Abstract*—In Forex market, designing effective strategies are a critical role in investment. However, it is a challenging task due to its inherent characteristics, which include high volatility, trend, noise, and market shocks. In this paper, we propose four actor-critic-based algorithms: Proximal Policy Optimization (PPO), Actor-Critic using Kronecker-Factored Trust Region (ACKTR), Deep Deterministic Policy Gradient (DDPG), and Twin Delayed DDPG (TD3). It employs deep reinforcement schemes to learn a stock trading strategy by maximizing investment return. Besides, the ensemble trading strategy was combined with four algorithms to find the best method. We use our algorithms to test the 30 forex currencies.**

***Index Terms*—Forex Trading, Deep Reinforcement Learning, TD3, PPO, DDPG, Automation Trading, ACKTR, Actor-Critic**

I. Introduction

The forex (FX) marketplace is one of the most popular monetary markets for buying and selling currencies because it is the largest financial marketplace inside the global in terms of trading volume [1]. The main players in the FX market are central banks, commercial banks, investment companies, hedge corporations, and those who also are regarded as investors. Their goal of participation in the FX market comes from their financial activities and economic needs. Large participants mostly engage in the FX market to manage their portfolios and avoid the exchange-rate risk. Small participants usually play in the FX market to make a profit from the short-term currency rate changes. To put it simply, both participants aim to define a profitable trading strategy. [2] However, it is challenging for analysis to consider all relevant factors in a complex and dynamic FX market.

We approach for FX trading is to model it as a Markov Decision Process (MDP) and use dynamic programming to derive the optimal strategy. However, the scalability of this model is limited due to the large state spaces when dealing with the FX market [3].

In recent years, machine learning and deep learning algorithms have been widely applied to build prediction and classification models for the financial market. Nevertheless, the disadvantage of these algorithms that are not trained to model positions [4]

In this paper, we propose four deep reinforcement learning algorithms and the ensemble strategy that combines ones that finds the optimal trading strategy in a complex and dynamic stock market. The four actor-critic algorithms are Proximal Policy Optimization (PPO) [5] [6] [12], Actor-Critic using Kronecker-Factored Trust Region (ACKTR) [7], [8], Deep Deterministic Policy Gradient (DDPG) [13], and Twin Delayed DDPG (TD3) [10] [11].

II. Related Works

A. Modelization

We model the forex trading process as a Markov Decision Process (MDP):

- State $s = [p, h, b]$: a vector that includes forex prices $p \in R_+^D$, the amount of foreign curennncy holding $h \in Z_+^D$ and the remaining balance $b \in R_+$, where D denotes the number of forex categories and Z_+ denotes non-negative intergers
- Action a: denote three actions buy, sell, hold currency
- Rewarding $r(s, a, s')$: the direct reward of taking action a at state s and arriving at the new state s'
- Policy $\pi(s)$: the trading strategy at state s, which is probability distribution of actions at state s
- Q-value $Q_(\pi)(s, a)$: the expected reward of taking action a at state s following policy π

B. The state transition of a forex trading process

- Selling: $k[d] \in [1, h[d]]$, where $d = 1, \ldots, D$ currency pairs can be sold from the current holdings, where k must be an integer. In this case, $h_{t+1}[d] = h_t[d] - k[d]$
- Holding: $h_{t+1}[d] = h_t[d]$
- Buying: $k[d][1, h[d]]$, where $d = 1, \ldots, D$ currency pairs can be bought and it leads to $h_{t+1}[d] = h_t[d] + k[d]$. In this case $a_t[d] = -k[d]$ is a negative integer.

C. Integrating Forex Trading Constraints

- Transaction cost: there are multiple types of transaction costs such as exchange fees, execution fees, SEC fee and so on. Because of numerous fees, so

978-1-6654-6166-5/22/$31.00 ©2022 IEEE

we just assume our transaction costs to be 0.1% of the value of each trade.

- Risk-aversion for market crash: the forex market can always be volatile, such as inflation, wars, financial crisis. To control the risk in a worst-case scenario like 2008 global financial crisis or economic recession after the covid pandemic and the conflict between Russia and Ukraine in 2022. We apply on the financial turbulence index $turbulence_t$ defined in Eq. (1) to partially make market forecasts to improve model efficiency [9]:

$$turbulence_t = (y_t - \mu)\Sigma^{-1}(y_t - \mu)', \qquad (1)$$

where $y_t \in R^D$ denote the forex return of current period t, μR^D denote the average of historical return

D. Return Maximization as Trading Goal

A reward function was design base on the balance b of the account when making a transaction. The goal is to design a strategy to maximize the value of the reward function (defined in Eq. (2))

$$r(s_t, a_t, s_{t+1}) = b_{t+1} - b_t \qquad (2)$$

With multiple transaction, we define our reward function as the change of the portfolio value when action a_t is taken at state s_t and arriving at new state s_{t+1}. It can be rewritten as Eq. (3)

$$r(s_t, a_t, s_{t+1}) = r_H + r_S + r_B - c_t \qquad (3)$$

Where r_H (defined in Eq. 4) ,r_S (defined in Eq. 5) and r_B (defined in Eq. 6) denote the change of the portfolio value comes from holding, selling, and buying currency from time t to $t+1$, c_t is transaction cost.

$$r_H = (p^H_{t+1} - p^H_t)^T k_t \qquad (4)$$

$$r_S = (p^S_{t+1} - p^S_t)^T k_t \qquad (5)$$

$$r_B = (p^B_{t+1} - p^B_t)^T k_t \qquad (6)$$

Turbulence index $turbulence_t$ is incorporated with the reward function to address our risk-aversion for marketing cash. When the index (1) goes above a threshold. Equation (5) becomes (7).

$$r_s = (p_{t+1} - p_t)^T k_t \qquad (7)$$

The model is initialized as follows. p_0 is set to the forex prices at time 0 and b_0 is the number of initial funds. The h and $Q_\pi(s,a)$ are 0, and $\pi(s)$ is uniformly distributed among all actions for each state. Then, $Q_\pi(s_t, a_t)$ is updated through interacting with the forex environment. The optimal strategy is given by the Bellman Equation, such that the expected reward of taking action a_t at state s_t is the expectation of the summation of the direct reward $r(s_t, a_t, s_(t+1))$ and the future reward in the next state $s_(t+1)$. Let the future rewards be discounted by a factor of $0 < \gamma < 1$ for convergence purpose, then we have

$$\begin{aligned} Q_\pi(s_t, a_t) = E_{s_{t+1}}[r(s_t, a_t, s_{t+1}) + \\ \gamma E_{a_{t+1}]\sim\pi(s_t+1)}[Q_\pi(s_{s+1}, a_{t+1}]] \end{aligned} \qquad (8)$$

We employ the deep reinforcement learning method to design a trading strategy that maximizes the positive cumulative change of the portfolio value $r(s_t, a_t, s_{t+1})$ in the dynamic environment. [4]

III. Set up the Forex trading environment

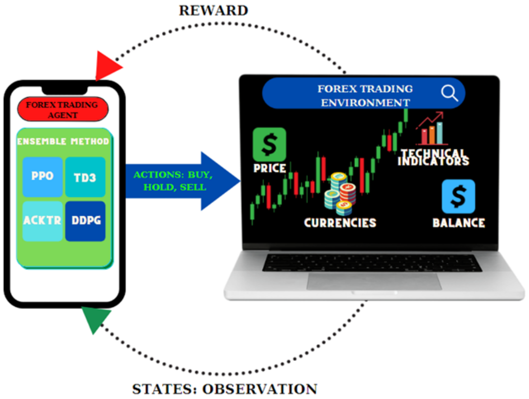

Fig. 1: Overview of the proposed system

The overview of the proposed system is illustrated in Fig. 1. Building a Forex trading environment should have a lot of information to use such as the amount of currency holding, historical price data, and information about technical indicators. To be able to put the information into the environment, you can deploy the environment using OpenAI gym to easily add information and train the model

A. State space

To represent the state space of multiple currency pairs environment, we use a vector in 211 dimensions: $[b_t, p_t, h_t, M_t, MS_t, MH_t, R_t, V_t]$.

We define each element as:

- $b_t \in R_(+)$: available balance at current time step t
- $p_t \in R^{30}_+$: close price of each forex pair
- $h_t \in R^{30}_+$: the volume owned by each pair
- $M_t \in R^{30}_+$: the MACD line of the MACD indicator
- $MS_t \in R^{30}_+$: the Signal line of the MACD indicator
- $MH_t \in R^{30}_+$: the Histogram line of the MACD indicator
- $R_t \in R^{30}_+$: the RSI line
- $V_t \in R^{30}_+$: the VWMA indicator line

B. Indicators

1) Moving Average Convergence/Divergence (MACD) indicator:

- Developed by Gerald Appel, the Moving Average Convergence/Divergence oscillator (MACD) [14] is one of the simplest and most effective momentum indicators available. The MACD indicator includes 3 lines: The MACD line, the Signal line, and the Histogram line. The exponential moving average (EMA) [20] is a type of moving average, EMA uses the value of $n - period$. In this paper, we choose the value of a period as 5 minutes. The formula of EMA is calculated as in Eq. (9):

$$EMA_t = kP_t + (1-k)(EMA_{t-1}) \tag{9}$$

  P_t is calculated using the close price of a currency pairs

  The smoothing parameter k takes on a value of between 0 and 1, typically chosen as $\frac{2}{m+1}$. An example is shown below for the computation of EMA where m = 9 (periods) and therefore $k = \frac{1}{5}$
- The MACD line is the difference between two exponential moving average (EMA), the most commonly used values are EMA 12 and EMA 26 (defined in Eq. (10)):

$$M_t = 12 - periodEMA_t - 26 - periodEMA_t \tag{10}$$

- The signal line of the MACD is usually a 9-period EMA (defined in Eq. (11)), which acts as a signal line and identifies turns

$$MS_t = 9 - periodEMA_t \tag{11}$$

- The MACD Histogram line represents the difference between the MACD line and its $9 - period$ EMA (defined in Eq.(12)).

$$MH_t = M_t - MS_t \tag{12}$$

2) Relative Strength Index (RSI) indicator:

- The Relative Strength Index (RSI) [15] is calculated using closing prices. RSI quantifies recent price changes. If the price moves around the support line, it indicates the stock is oversold and we can take buy action. If the price moves around the resistance, it indicates the stock is overbought and we can take selling action
- The RSI uses a two-part calculation that starts with the Eq. (13):

$$RSI_{stepone} = 100 - [\frac{100}{1 + \frac{Average\_gain}{Average\_lost}}] \tag{13}$$

  Because $RSI_{stepone}$ uses historical data of the previous period, the RSI formula in the first period will be different from the following periods. The formula of later RSI step is calculated as in Eq (14).

$$RSI_{later} = 100 - [\frac{100}{1 + \frac{PreviousAverage\_gain * 13}{PreviousAverage\_lost * 13}}] \tag{14}$$

- Traders often use the 20 and 70 values to trade with the RSI line. When the RSI line falls below 20, which shows an oversold signal, traders should be inclined to buy action. Conversely, when the RSI [15] value exceeds the value of 70 will be a sign of overbought, and traders should be inclined to take profit selling action.

3) Volume-Weighted Moving Average(VWMA):

- Volume value plays an important role in making forex trading decisions. Volume values can either reflect the interest of a large number of traders in a price zone or be a sign of a market maker taking action. Using the VWMA [16] indicator will assist in making more accurate decisions. The formula of VWMA for n-period is calculated as in Eq. (15).

$$V_t = \frac{\sum_{i=0}^{n}(V+P)}{\sum_{i=0}^{n}(V)} \tag{15}$$

  where V is the volume of the period, P (defined in Eq. (16)) is the Average Price from the opening bell. Using the highest price h, lowest price l, and close price c.

$$P = \frac{h+l+c}{3}. \tag{16}$$

  The VWMA line is often used in combination with other lines such as RSI, and SMA [17] to help increase the accuracy of market analysis.

C. Action space

- In forex trading, because the value of currencies is usually kept stable, the price change is usually not high. To achieve greater profits, traders need to use leverage to trade such as 1:100, 1:500. And the concept of a lot in forex usually represents 100,000 trading units. This means that if you want to trade the dollar with a volume of 1 lot, you must have 100,000 USD as a trading reference. However, there are currently many lot sizes that investors can choose from $Mini$, $Micro$, and $Nano$ Lot with 10,000, 1,000, and 100 currencies respectively. However, for the scope of this article, we will use the normal amount instead of the lot index.
- Action space is defined as {-n, ...,- 1, 0, 1, ..., +n}. When the value of action space is negative, it represents the action of selling k products, whereas when the value of action space is positive, it represents the action of buying k products, when the value of action space is 0 it represents the action of holding.

IV. Trading Agent base on Deep Reinforcement Learning

We use four actor-critic based algorithms to implement our trading agent. These algorithms are PPO [5], [6], [12] , DDPG [13], TD3 [10] [11] and ACKTR [7], [8]. We apply these algorithms respectively for each agent. We also use an ensemble strategy which is proposed to combine the four agents together to build a robust trading strategy.

A. Proximal Policy Optimization (PPO)

- PPO [5] [6] [12] is the algorithm developed by OpenAI. PPO algorithm combines the idea of the A2C [18] algorithm (having multiple workers) and TRPO [19] algorithm (Using a trust region to improve the actor). Assuming that the probability ratio between old and new policies is expressed as (17):

$$r_t(\theta) = \frac{\pi_\theta(a_t, s_t)}{\pi_{\theta_{old}}(a_t, s_t)} \tag{17}$$

- The main idea of PPO is to discourage large policy changes to move outside of the clipped interval by introducing a clipping term to the objective function (defined in Eq. (18)).

$$L_\theta^{CLIP} = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{18}$$

  where $r_t(\theta)\hat{A}_t$ is the normal policy gradient objective of state s and action a at time t, r_t s the ratio of the probability between the new and old policies, $\hat{E}_t$ denotes the empirical expectation over timesteps, $\hat{A}_t$ is the estimated advantage at time t. The function $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)$ clips the ratio $r_t(\theta)$ to be within range $[1-\epsilon, 1+\epsilon]$.
- PPO is the off-policy algorithm. This algorithm uses important trading historical samples and has more advantages in such cases as lack of historical data.

B. Actor-Critic using Kronecker-Factored Trust Region(ACKTR)

- Actor-Critic using Kronecker-Factored Trust Region (ACKTR) [7] [8] which was developed by researchers at the University of Toronto and New York University combines three distinct techniques: Trust Region Optimization (for more consistent improvement), distributed Kronecker factorization, and actor-critic methods to improve sample scalability and efficiency.
- ACKTR uses Kronecker-factored approximated curvature (K-FAC) in the natural policy gradient to efficiently approximate Fisher information matrices and inverse matrices. Fisher information matrix is defined in Eq. (19) :

$$F = E_{p(r)}[(\bigtriangledown_\theta log\pi(a_t|s_t))(\bigtriangledown_\theta log\pi(a_t|s_t)^T)] \tag{19}$$

  Parameter $pi(a_t|s_t)$ is the policy that decides at with the condition st; θ indicates weights of neural network for policy, where θ decides the policy π .In this application, a neural network is used, and the actor-critic model may be called neural actor-critic [3]. $\bigtriangledown_\theta$ is the gradient descent of θ; $p(\tau)$ is the trajectory distribution.
- We choose the ACKTR model as an agent because of its stability and its efficiency. Therefore, it is a great model for Forex trading.

C. Deep Deterministic Policy Gradient (DDPG)

- DDPG [13] is the combination of Q-learning and policy gradient using neural networks as function approximators.
- At each time step, DDPG performs an action a_t at s_t, receives a reward r_t and comes to state s_{t+1}. The replay buffer R stores the transitions (s_t, a_t, s_{t+1}, r_t). A batch of N transitions is drawn from the replay buffer R. The Q-value y_i is updated as Eq. (20) :

$$y_i = r_i + \gamma Q^{'}(s_{i+1}, \mu^{'}(s_{i+1}|\theta^{\mu^{'}}, \theta^{Q^{'}}). \tag{20}$$

- Critic network is then updated by minimizing the loss function L(θ^Q) (defined in Eq. (21)) which is the expected difference between outputs of the target critic network Q and the critic network Q'.

$$L(\theta^Q) = s_{(s_t,a_t,R_t,s_{t+1})\sim buffer}[(y_i - (s_t, a_t|\theta^Q))^2] \tag{21}$$

- We choose DDPG algorithm because it is effective when handling with continuous action space. Therefore, it is appropriate for the forex trading environment.

D. Twin Delayed DDPG (TD3)

- TD3 [10] [11] is the improved algorithm of DDPG [13]. Although DDPG sometimes can have high performance, it is frequently brittle when faced with hyperparameters and other kinds of tuning. Twin Delayed DDPG (TD3) is an algorithm that solves the overestimation bias of DDPG by having three critical improvements [10] (Clipped Double-Q Learning, "Delayed" Policy Updates, and Target Policy Smoothing).
- The first improvement is "Target Policy Smoothing". (defined in Eq. 22) Deterministic policy methods tend to produce target values with high variance when updating the critic. This is caused by overfitting spikes in the value estimate. Actions that are based on the target policy (µtarget) are used to update the Q-learning target, then will be added clipped noise on each dimension of the action. Next, the target action is clipped to lie in the valid action range (all valid actions and satisfy $a_{low} < a < a_{high}$)

$$a'(s') = clip(\mu_{\theta target} + clip(\epsilon, -c, c), a_{low}, a_{high}), \epsilon \in N(0, \sigma) \tag{22}$$

- The second improvement of TD3 is ”Clipped Double-Q Learning” (defined in Eq. 23). TD3 learns two Q-functions instead of one and uses the smaller of them to form the targets in the Bellman error loss functions.

$$y(r, s', d) = r + \gamma(1-d)\underset{i=1,2}{min} Q_{\Phi_{i,target}}(s', a'(s')), \tag{23}$$

then both of them are learned by regressing to the target (Eq. (24) and Eq. (25)) .

$$L(\Phi_1, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{E}[(Q_{\Phi_1}(s,a) - y(r, s', d))^2] \tag{24}$$

$$L(\Phi_2, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{E}[(Q_{\Phi_2}(s,a) - y(r, s', d))^2] \tag{25}$$

- The third improvement of TD3 is ”Delayed Policy Updates” (defined in Eq. (26)). Because TD3 uses the smaller Q-value and regresses towards the target so it reduces the chance of overestimation in the Q-function. We can see the ”Delay” in updating the policy. The policy is learned by maximizing Q_{Φ_1}

$$\underset{\theta}{max} \underset{s\sim\mathcal{D}}{E}[Q_{\Phi_1}(s, \mu_\theta)(s)] \tag{26}$$

- We can see that TD3 updates the policy less frequently than DDPG. We choose TD3 because we want to see the difference between TD3 and DDPG in the automation Forex Trading problem.

E. Ensemble Strategy

We combined 4 Deep Reinforcement Learning models: TD3, PPO, DDPG, and ACKTR. In each period of trading time, this strategy chooses the model which has the highest Sharpe Ratio. Sharpe Ratio which is developed by William F. Sharpe is used for helping traders know the return of an investment compared to its risk.

- Step 1: Let the model trade in the validation set and trading set in the chosen period (50 days).
- Step 2: In each period, we calculated the value of the Sharpe ratio (Eq. (27)) of each model.

$$Sharpe\,ratio = \frac{r_p - r_f}{\sigma_p}, \tag{27}$$

where r_f is the risk-free rate, r_p is the return of portfolio, σ_p is the standard deviation of the portfolio’s excess return.

- Step 3: After having 4 Sharpe ratios of 4 models, this strategy will choose the model which had the highest Sharpe ratio to trade in the next period.

V. Implementation Process

A. Data collection and Preprocessing

- Data Source: FXCM, also known as Forex Capital Markets, is a retail forex broker for trading in the foreign exchange market. FXCM allows anyone to speculate in the foreign exchange market and offers trading in contracts for difference (CFDs) on major indices and commodities such as gold and crude oil. It is based in London.
- FXCM provides a RESTful API to interact with its trading platform. Among others, it allows retrieving historical data as well as streaming data. In addition, it allows to place different types of orders and read account information. The overall goal is to enable the implementation of automated, rhythmic trading programs
- Data collection and Preprocessing **steps** :
  1) Install the necessary libraries and packages.
  2) Go to https://tradingstation.fxcm.com/ and get an API token to collect data through fxcmpy – a Python library
  3) Select 30 most popular currency trading pairs on the exchange to collect, collection time from 02/01/2018 to 01/07/2022, time interval, the collection time interval is 1-day candle.
  4) Handle other operations with pandas: convert date data into integer form of the form %Y%d%m, bid-ask values are auction ranges so the group chooses the average value as the final value.
  5) Add columns of indicator values, and turbulence. stockstats is a python library that helps calculate the value of popular indicators today. Use the stockstats library to add value columns of indicators, volume,... .

B. Data Splitting for Training

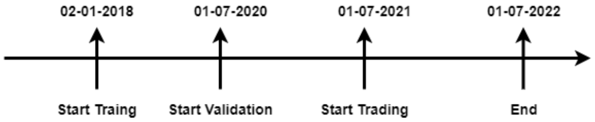

Fig. 2: Split dataset by time points

The dataset is divided into 3 parts: training, validation and trading (shown in Fig. 2). Training from January 2, 2018 to June 30, 2020, then the system will conduct trade validation to evaluate the algorithms. calculations, models for the period from July 1, 2020 to June 30, 2021. After that, the system will automatically go for real trading from July 1, 2021 to July 1, 2022.

C. Experimental results and evaluation

Overall, all of the aforementioned methods bring good results (profits) for the account of investors at the end of the period, all 5 methods bring profits 15% larger than the initial amount (1000000 USD) to investors

The model with the best result in the final stage is the ACKTR model which brings investors a profit of nearly 12.5% on the last day. The models that give the lowest results of all 5 models in this period are the DDPG model and the Ensemble method (at this stage, DDPG is chosen as the model to follow by ensemble strategy).

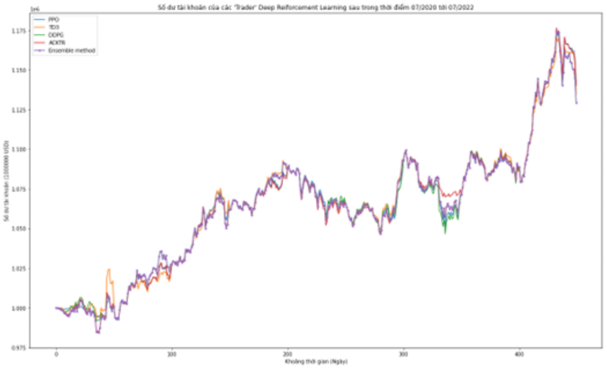

Fig. 3: Results (Account Balance) of algorithms

Figure 3 show us the comparison about the account balance change among all trading agents. The final stage (from the 400th-day mark onward) is the period where the profit fluctuates the most when simultaneously increasing from 10.75% to more than 17.5% and decreasing after that. Most of the time, the methods bring acceptable profit to the investor. However, in the first period of trading processes, five methods applied for the agent did not bring good results for the investors. The middle period (between 150th day and 300th day) is the period when the investor's account balance sees the least volatility as the investor's account balance always fluctuates in the range of 5% to 10%. The ensemble method works quite well most of the time in the trading process, but in the last period, it did not give the best result in the final period.

VI. Conclusion

The foreign exchange market (Forex, FX, or money market) is a global decentralized market for the exchange of currencies. With the forex price fluctuating constantly, the design of a good investment strategy plays an important role in the success of investing in this market.

Deep Reinforcement Learning algorithms when entering this market give us a temporarily acceptable result when all 5 methods yield profitable results (more than 15%) for investors after 450 days of trading.

Acknowledgment

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

References

[1] Y. W. Cheung and M. D. Chinn, "Currency traders and exchange rate dynamics: A survey of the US market," J. Int. Money Finance, vol. 20, no. 4, pp. 71–439, Aug. 2001.

[2] Munkhdalai, Lkhagvadorj Munkhdalai, Tsendsuren Park, Kwang Ho Lee, Heon Li, Meijing & Ryu, Keun. (2019). Mixture of Activation Functions With Extended Min-Max Normalization for Forex Market Prediction. IEEE Access. PP.1-1.10.1109/ACCESS.2019.2959789.

[3] Brown, B., & Zai, A. (2020)."Deep reinforcement learning in action". Manning Publications Co.

[4] Yang, Hongyang Liu, Xiao-Yang Zhong, Shan Walid, Anwar. (2020). Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. SSRN Electronic Journal. 10.2139/ssrn.3690996.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347, 07 2017.

[6] Zhipeng Liang, Kangkang Jiang, Hao Chen, Junhao Zhu, and Yanran Li, "Adversarial deep reinforcement learning in portfolio management," arXiv: Portfolio Management, 2018.

[7] Yuhuai Wu, Elman Mansimov, Shun Liao, Roger Grosse, Jimmy Ba, "Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation", arXiv:1708.05144

[8] F. Heryanto, B. D. Handari, and G. F. Hertono , "Trading financial assets with actor critic using Kronecker-factored trust region (ACKTR)", AIP Conference Proceedings 2242, 030001 (2020) https://doi.org/10.1063/5.0008090

[9] Mark Kritzman and Yuanzhen Li, "Skulls, financial turbulence, and risk management," Financial Analysts Journal, vol. 66, 10 2010.

[10] Twin Delayed DDPG. (n.d.). OpenAI Spinning Up. Retrieved August 17, 2022, from https://spinningup.openai.com/en/latest/algorithms/td3.html

[11] Scott Fujimoto, Herke van Hoof, David Meger, "Addressing Function Approximation Error in Actor-Critic Methods", arXiv:1802.09477

[12] John Schulman, Oleg Klimov, Filip Wolski, Prafulla Dhariwal and Alec Radford, "Proximal Policy Optimization". OpenAI. Retrieved August 17, 2022, from https://openai.com/blog/openai-baselines-ppo/

[13] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra, "Continuous control with deep reinforcement learning", arXiv:1509.02971

[14] Jason Fernando,Moving Average Convergence Divergence (MACD),Investopia, Retrieved August 17, 2022, 2022, from https://www.investopedia.com/terms/m/macd.asp

[15] Jason Fernando,Relative Strength Index (RSI),Investopia, Retrieved August 17, 2022, from https://www.investopedia.com/terms/r/rsi.asp

[16] Galen Woods, Volume-Weighted Moving Average (VWMA) – A simple volume tool, Trading Setups review, Retrieved August 17, 2022 , from https://www.tradingsetupsreview.com/volume-weighted-moving-average-vwma/

[17] Adam Hayes, Simple Moving Average (SMA), Investopia, Retrieved August 17, 2022, from https://www.investopedia.com/terms/s/sma.asp

[18] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, Koray Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning", arXiv:1602.01783

[19] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, Pieter Abbeel. "Trust Region Policy Optimization", arXiv:1502.05477

[20] James Chen, Exponential Moving Average (EMA), Investopia, Retrieved August 17, 2022, from https://www.investopedia.com/terms/e/ema.asp