TraderNet-CR: Cryptocurrency Trading with Deep Reinforcement Learning

Vasilis Kochliaridis[1][0000−0001−9431−6679], Eleftherios Kouloumpris[1][0000−0003−1214−3845], and Ioannis Vlahavas[1][0000−0003−3477−8825]

School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

Abstract. The predominant method of developing trading strategies is technical analysis on historical market data. Other financial analysts monitor the public activity towards cryptocurrencies, in order to forecast upcoming trends in the market. Until now, the best cryptocurrency trading models rely solely on one of the two methodologies and attempt to maximize their profits, while disregarding the trading risk. In this paper, we present a new machine learning approach, named TraderNet-CR, which is based on deep reinforcement learning. TraderNet-CR combines both methodologies in order to detect profitable round trips in the cryptocurrency market and maximize a trader's profits. Additionally, we have added an extension method, named N-Consecutive Actions, which examines the model's previous actions, before suggesting a new action. This method is complementary to the model's training and can be fruitfully combined, in order to further decrease the trading risk. Our experiments show that our model can properly forecast profitable round trips, despite high market commission fees.

Keywords: Cryptocurreny Trading · Deep Reinforcement Learning · Public Activity Analysis · Technical Analysis · Risk Management

1 Introduction

Cryptocurrency tokens have become particularly interesting trading assets, due to their high volatility [12]. Many professional investors and financial analysts are turning to technical analysis, in order to estimate the future prices of cryptocurrencies and spot trading opportunities. Unlike fundamental analysis, which requires a company's financial position, technical analysis merely requires a mathematical formula to be applied to prior market data. Technical analysis provides pattern-based indicators of the momentum, volatility and trend of an asset. [14].

Algorithmic trading i.e. the use of computer programs to automate quantitative trading methods, is an essential step towards a more exact specification and implementation of technical analysis. Although algorithmic trading is beneficial due to the speed with which orders are executed, it is primarily reliant on technical indicators, which are prone to producing false buy/sell signals and market trends. To overcome this issue, traders consider combinations of indicators, however it has yet to be determined which combinations are the most effective in each circumstance.

Various studies attempted to apply machine learning techniques to cryptocurrency trading, based on algorithmic trading, as detailed in more depth in the related works Section. The same works prove that especially Deep Reinforcement Learning (DRL), which is a sub-field of machine learning, has the potential to outperform traditional trading strategies. However, our research revealed that past studies have overlooked three critical characteristics of cryptocurrency trading. Firstly, numerous popular and widely used indicators were missing from the training data. They also lack a public activity index which, as we prove later in the paper, contains valuable information about the prices. Finally, several previous models that aim to optimize a portfolio's wealth disregard the trading risk, which is an important aspect of a trading strategy, as also highlighted in section 2.

In this paper we present TraderNet-CR, a DRL agent [1] which relies on both technical analysis and hourly public activity towards cryptocurrency assets. Our agent's actions are intended to exploit potentially beneficial round trips in a market [2] with low risk. The remaining of the paper is structured as follows. Section 2 describes the related work, while Section 3 describes the methodological framework in detailed steps. Then, Section 4 discusses the empirical results and finally, Section 5 concludes this study and presents future research avenues and possible improvements of the algorithm.

2 Related Work

In this Section, we exclusively review works that apply DRL to find optimal trading strategies in a cryptocurrency market. Satarov et al. [16] applied the Deep Q-Learning (DQN) algorithm in order to identify profitable trading points. In this work, their agent was rewarded only during sell actions, with the reward being a subtraction between the current selling price and the most recent buying price. In addition, penalties were given to the same sequential actions. Considering trading fees of 0.15 percent, the work demonstrated that the Reinforcement Learning (RL) approach performed better than three traditional technical strategies.

Jiang et al. [8] formulated a multi-asset portfolio management problem of high-volumed cryptocurrencies, with a DRL setting that was implemented for both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) and parameter sharing between different assets. The external state is represented as a tensor of historical price ratios for every considered asset. The internal state includes the portfolio weight vector that specifies the current allocation of capital, has length equal to the considered assets and a total sum equal to 1. The immediate rewards of their agent are expressed as the 1-period logarithmic return of the portfolio. Commission fees of 0.25 percent are integrated with

[1] A DRL agent utilizes a deep learning model in order to learn to behave optimally in its environment.

[2] A round trip is a pair of two opposite orders placed one after the other (buy-sell or sell-buy), that aims to take advantage of price differences in order to produce profit.

the introduction of a penalty analogous to the change in the portfolio weights. [8]

While the previous work considers a single RL agent to manage the entire portfolio, Lucarreli and Borrotti [11] employed a multi-agent framework by training local RL agents for each financial asset (Bitcoin, Etherium, LiteCoin, Riple). The performance of the each local agent produced a local reward signal, which is combined with the rest signals to formulate a global reward signal. The goal of this multi-agent framework was the maximization of the global reward signal, in order to achieve optimal portfolio management. The state space consisted of closing prices across all assets. Even though they achieved very promising results, they completely disregarded the commissions fees.

To finish with this short related work review, a major problem of the existing literature is that the current state of the art DRL methodologies operate on low commission fees. Additionally, in their works, they prioritized in maximizing the investment profits, rather than minimizing its risk. In our work, we aim to improve upon existing literature by (a) including new features such as technical indicators and public activity indicators, (b) experimenting with a more advanced deep RL algorithm design, (c) adding a trading rule as an extension of our main algorithm, which customizes the agent's trading behavior and further reduces the trading risk.

3 Methodology

Cryptocurrency trading poses several challenges for reinforcement learning for various reasons. First of all, since the cryptocurrency market involves non-stationary and noisy time series data, the prediction of future prices and directional movements becomes a quite difficult task [9]. Additionally, an RL agent will make a sequence of actions in order to maximize its rewards, however it is hard to reward that sequence of actions before the end of the evaluation period, which often leads to the sparse rewards problem [3]. In this Section, we discuss some of the methods that are used to tackle the above challenges in cryptocurrency trading. Additionally, we propose a new method, named N-Consecutive actions method, which is used to further reduce the trading risk.

3.1 Problem Formulation

Unlike prior efforts, we omit the portfolio's wealth from the agent's input state to simplify the complexity of the stochastic nature of trading. Rather than attempting to maximize its initial wealth, the agent is trained to earn profits by spotting profitable round trips and taking the appropriate hourly actions. This is accomplished by utilizing a reward function that compensates the agent's actions based on the maximum future profit they may generate, as described in more

[3] The sparse reward problem happens when an environment rarely produces a reward. This usually slows down the training process of a DRL agent. [15]

detail later in this Section. There are three available actions to our agent. At each timestep, our agent may either suggest to BUY or SELL a unit or HOLD.
State Space. Our state space is represented with a matrix $s = [c, v, t, g, d]$ of S-dimensional columns, where S denotes a timeframe size, which is used to define the number of previous feature rows that are included in the current state. In our experiments, we found that $S = 20$ is an ideal timeframe size for every cryptocurrency asset. Each row represents the state of a time step (the state in a previous hour) as a vector $s_0, s_1, ..., s_{18}$, with s_{19} as the current state. The state includes the close differences $c \in R$ of consecutive hours, the volume differences $v \in R$, the 24-hour time index $t \in [0, 23]$, the google trends score $g \in [0, 100]$, and the technical indicators, based on the past data $d \in R^D$.
Public Activity. Public activity may occasionally foreshadow impending bullish or bearish signals [4]. We define as public activity the time of the day which the trading takes place, as well as the google trends score in that specific hour. Google trends, is a 0 to 100 scale that measures the online traffic of searched terms. The terms that we used in our experiments were the names of the cryptocurrency assets. This indication could be highly valuable in cases where the online presence of influencers causes unexpected spikes or drops of the prices and volumes.
Technical Indicators. At each state, we compute the technical indicators using prior market data. The indicators are listed as follows:

- **Exponential Moving Average** (EMA): a moving average indicator that was serves as a building block for several other indicators [3].
- **Double Exponential Moving Average** ($DEMA$): a moving average indicator that is used to reduce market noise in price charts. Unlike EMA, it contains less lag and it is consider more responsive. [13].
- **Moving Average Convergence Divergence** ($MACD$): a trend indicator that compares the the EMAs of two different windows. [5].
- **Volume-Weighted Average Pricing** ($VWAP$): a weighted average technical indicator that is computed by adding up the close price for every transaction, mainly used by financial institutions and funds. [3].
- **Relative Strength Index** (RSI): a momentum indicator that measures the magnitude of recent price changes to assess overbought or oversold conditions.[4].
- **Intraday Momentum Index (*IMI*)**: an alternative indicator to RSI that considers the relationship between the opening and thec losing price over the course of the day[10].
- **Average Directional Index** (ADX): a trend strength indicator that is bounded between 0 and 100, just like RSI and IMI [4].
- **Commodity Channel Index** (CCI): an indicator which can gauge an overvalued or undervalued market. In contrast to other oscillators that range in a bounded interval [1].

[4] A signal is called bullish when the close price begins to rise. On the other hand, a signal is called bearish when the close price starts to drop.

- **On-balance volume** (OBV): a momentum indicator that relies on patterns of volume flow to predict changes in price. [7].
- **Accumulation/Distribution Indicator** (A/D): an indicator which can estimate if volume flow is adequate for the continuation of a trend, or whether a reversal is about to take place [6].
- **Bollinger Bands** ($BBands$): It is a technical analysis tool that defines a interval specified by adding and subtracting 2 moving standard deviations from a Simple Moving Average (SMA) signal [2].

Architecture. We selected the Proximal Policy Optimization (PPO) algorithm as the agent's architecture, because it is fast, stable and has been proven to achieve state of the art results in many RL environments. For the actor network, we used a convolutional neural network to represent the policy. The convolutional layer with 32 filters, kernel size of 5 and stride of 1, which ends up in a fully connected network. The fully connected network includes two hidden layers of 256 units each and relu activation functions. The same architecture was used for the critic network, which uses the Adam optimizer to update its weights with Learning Rate $Lr = 0.00025$. As in the original paper [17], we set the clipping parameter $\epsilon = 0.3$, without parameter sharing between the two networks. We set each mini-batch of samples to be trained for 40 epochs. The architectures for the general PPO agent and TraderNet-CR actor-critic networks are provided in Figures 4 and 5 of Appendix A respectively.

Reward Function. In cryptocurrency trading, small increases or drops in the price of a cryptocurrency asset would result in unprofitable investments, due to high commission costs for each transaction. As it is quite improbable that the close price would change drastically during the first few hours of a transaction, the agent would have to wait many steps to determine whether an action that was suggested was correct be rewarded or penalized otherwise. This eventually leads to the sparse rewards problem. In order to detect profitable round trips, within the next k hours.

To address this issue, we designed the reward function in such a manner that the agent is rewarded based on the maximum return that an action can generate within the next K (hours). This eventually trains the internal layers of the agent's architecture to estimate the future price fluctuations within the near future. Given that f is the fee percentage, the reward function can be mathematically modeled as:

$$r_t = \begin{cases} C_{max} - C_t - f(C_{max} + C_t) & BUY \\ C_t - C_{min} - f(C_{min} + C_t) & SELL \\ -max(r_{t(a_i)}) & HOLD \end{cases} \tag{1}$$

with

$a_i \in \{BUY, SELL, HOLD\}$.

where

$$C_{max} = max\{C_{t+1}, C_{t+2}, ..., C_{t+k}\} \tag{2a}$$

$$C_{min} = min\{C_{t+1}, C_{t+2}, ..., C_{t+k}\} \tag{2b}$$

The above reward function ensures that if the agent anticipates a spectacular increase in the price when buying or a huge drop in the price when selling, then it receives a favourable reward. In cases where an action would lead to unintended losses, then it is penalized, in order to be discouraged of suggesting the same actions in similar states.

In many previous works, no reward was used ($r_t = 0$), during the holding time. However, in our investigation we have found out that the agent could sometimes prefer to converge to holding its position and avoiding any type of transaction, due to early negative returns. We encourage the agent to avoid holding, by penalizing it if it wrongly holds its position.

3.2 N-Consecutive Actions

Small price fluctuations in the market could possible distort the overall trend. Even with the use of technical analysis and public activity, the agent could be tricked by the market noise and suggest unprofitable actions. An indication of generating misleading actions could be in cases where the agent switches between BUY and SELL actions in consecutive timesteps. To avoid such cases, we defined a rule during the exploitation period, in which an action a_t will be accepted only if the N previously suggested actions by the agent are identical ($a_t = a_{t-1} = a_{t-2} = ... = a_{t-N}$). This method increases the probability that a generated action is profitable and thus reduces the trading risk. Furthermore, this method does not interfere with the agent's training and can be used as a safety mechanism that operates alongside with the agent's decision system.

4 Results

In this Section we analyze the importance of public activity and its correlation with the market data. Finally, we review the performance evaluation of the proposed approach. The experimental code supporting the results presented is publicly available and can be found on Github [5]. The training data consist of OHLCV [6] data from the last 5 years of six popular cryptocurrencies (Bitcoin, Ethereum, Solana, Cardano, Monero, Polygon) and were extracted from *Crypto Data Download* [7].

[5] https://anonymous.4open.science/r/Finance-AI-08C2

[6] OHLCV datasets consist of five columns: Open, High, Low, Close, Volume of a market at a specific time.

[7] https://www.cryptodatadownload.com/data/

4.1 Public Activity Importance

Throughout our research, we discovered that the *close* and *volume* features of our datasets are associated with the public activity. As shown in Plots (a), (b) of Figure 1, there are distinct hours during the day when the most transactions occur. The plot (1c) also shows that there are considerable fluctuations in the direction of the close price throughout the same hours. Also, it is worth mentioning that the trend scores also seem to be correlated with the time of day, as plotted in (1d). In order to correctly plot these correlations, we first standarized the data using a window of 24 hours and calculated the mean values of the features for each hour.

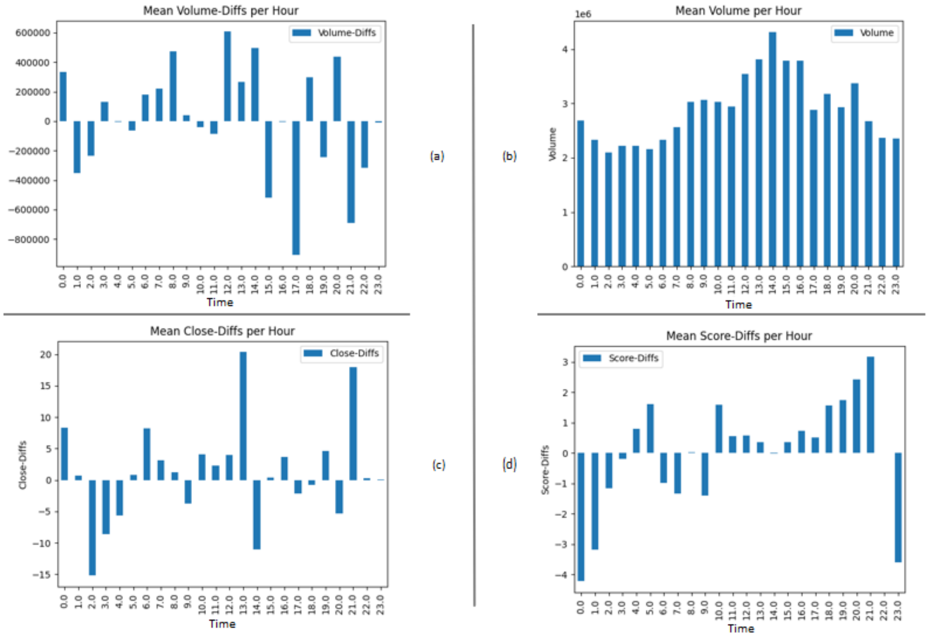

Fig. 1. Correlation analysis between the public activity and Bitcoin data

4.2 TraderNet-CR Evaluation

Our agent was trained separately in each market and was evaluated in the 15 latest successive days of the market dataset. The evaluation performance for each market can be seen in Figure 2. For each agent, we measured its mean rewards per hour, its theoretical maximum profit or loss (PNL) percentage and its risk at the end of the evaluation. The first metric measures the mean reward that the agent is receiving from the environment. The second metric measures the theoretical maximum profit percentage at the end of the evaluation period, if we always liquidate the shares that are generated by the agent's previous actions at the right time. Finally, the risk is defined as the percentage of the profitable transactions. To measure the agent's performance, we used commission fees of 0.5% and 1.0%. The Table 1 shows the performance of each agent for different commission fees.

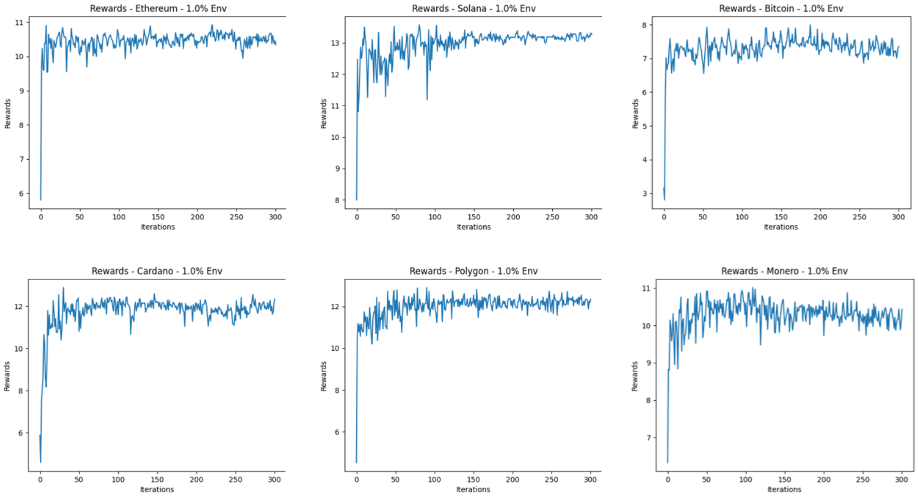

Fig. 2. The mean reward of the agent in each market. Commissions of 1.0 percent were used in each market.

Even though the Theoretical PNL indicates the maximum possible profit of the agent within a trading period, some traders might also be interested in the profit achievable with an actual trading strategy. In order to compute the profit, we used a trading strategy during the 15-day evaluation period, named Greedy PNL, in which we liquidated all the agent's shares generated by its previous actions, once they become profitable. Even though this strategy doesn't guarantee the maximum possible profit per round trip, it ensures that the agent doesn't miss profitable opportunities. The performance of this strategy can be show in Figure 3 and Table 1.

From our experiments, it is clear that our approach is profitable in every market that the agent was evaluated. In addition, we observe that the agent performed best in cryptocurrency markets with low trading volumes, such as Solana, Cardano and Polygon markets. To the best of our knowledge, these markets have not been included in previous DRL trading approaches.

Table 1. The metrics of the expertiments for each cryptocurrency asset.

Crypto Env	Mean Reward	Theoretical PNL %	Greedy PNL %	Risk %
Bitcoin 0.5%	10.12	248.96	2.6	0.15
Bitcoin 1.0%	7.36	15.84	0.63	0.26
Ethereum 0.5%	12.77	351.28	3.87	0.11
Ethereum 1.0%	9.42	123.91	0.97	0.2
Monero 0.5%	13.46	693.23	5.73	0.06
Monero 1.0%	10.43	339.52	1.24	0.18
Polygon 0.5%	13.46	702.71	7.83	0.06
Polygon 1.0%	10.43	336	2.83	0.18
Cardano 0.5%	15.71	662.63	33.69	0.04
Cardano 1.0%	12.73	229.79	7.47	0.17
Solana 0.5%	16.38	130.51	76.23	0.07
Solana 1.0%	13.27	58.72	19.58	0.16

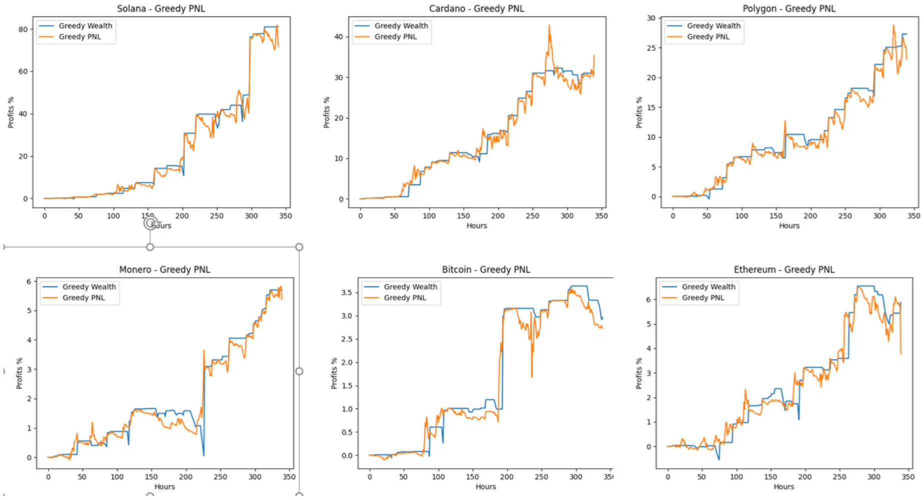

Fig. 3. The greedy PNL measurements for each experiment using the 2-Consecutive Actions rule

4.3 Optimizing Risk with N-Consecutive Actions

Every investor's principal aim is to generate as much profit as possible with the least amount of risk. However, some traders may prefer to trade only in situations where the likelihood of profiting from an investment is quite high. Using the "N-Consecutive Actions" rule, which is described in Section 3, we demonstrate how the risk can drop even further. As shown in Table 2, a decent window size of 2, can drastically decrease the trading risk in all markets. However, one should keep in mind that lowering an investment's risk may result in lower profit returns. This implies that the greater the window size, the lower the returns, but also the associated risks.

Table 2. The analytical trading risk for each agent, using window sizes ranging from 0 to 5. Zero length implies that no rule was used.

	Window Size (N)					
Crypto (1.0%)	$N = 0$	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$
Bitcoin	0.26	0.27	0.25	0.26	0.21	0.19
Ethereum	0.2	0.18	0.17	0.14	0.10	0.08
Monero	0.18	0.17	0.15	0.16	0.14	0.16
Polygon	0.18	0.15	0.13	0.14	0.12	0.12
Cardano	0.17	0.16	0.14	0.09	0.13	0.15
Solana	0.16	0.13	0.07	0.09	0.11	0.11

5 Conclusion

In this paper, we adopted a state of the art RL algorithm, named PPO, in order to detect profitable round trips with low trading risk. We used features from OHLCV market data, technical analysis and public activity indicators to represent the states of the environment. Additionally, we designed an intelligent reward function that boosts the agent's learning capability. After the training process, we applied the N-Consecutive Actions method, which increases the quality of the suggested actions. We tested our methodology in six popular cryptocurrencies for 15 successive evaluation days, using fees of 0.5% and 1.0%, in which the agent outputs an action every hour. Even with heavy commission fees and the most greedy liquidating strategy, the agent managed to deliver profits. In continuations of this work, we would like to investigate if portfolio wealth optimization can be improved using our methods, as well as add more rules to create a stronger end-to-end trading agent.

Appendix: A

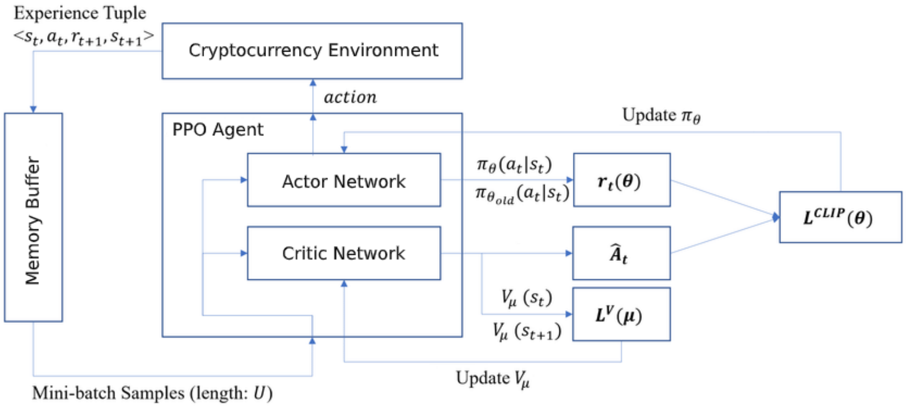

Fig. 4. A typical PPO Agent architecture

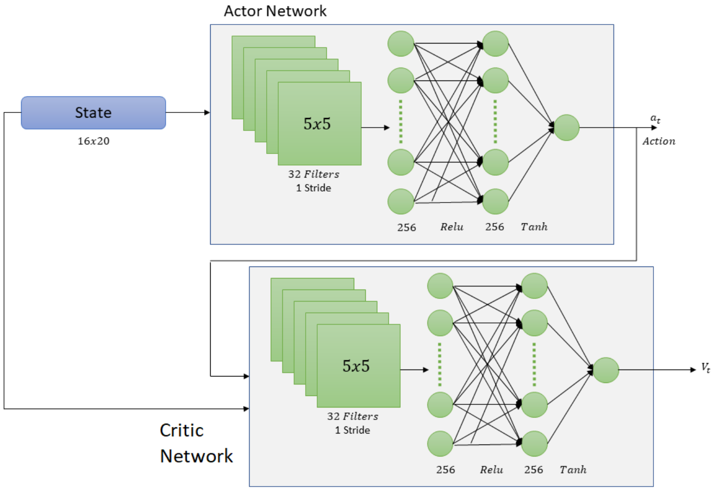

Fig. 5. The TraderNet-CR actor-critic network architecture

References

1. B., S.: The role of analyst forecasts in the momentum effect. Wiley Trading (2006)

2. Baiynd, A.M.: The trading book: A complete solution to mastering technical systems and trading psychology. McGraw-Hill (2011)
3. Brown, R.G.: Smoothing, Forecasting and Prediciton of Time Series. Dover Publications (1963)
4. Brown, R.G.: New Concepts in Technical Trading Systems. Trend Research (1978)
5. Brown, R.G.: Technical Analysis Power Tools for Active Investors. Financial Times Prentice Hall (2005)
6. Gerstein, M.: Evaluation of the chaikin power gauge stock rating system. Chaikin Analytics (2013)
7. Granville, J.E.: Granville's new key to stock market profits. Papamoa Press (2018)
8. Jiang, Z., Xu, D., Liang, J.: A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem pp. 1–31 (2017), `http://arxiv.org/abs/1706.10059`
9. Livieris, Ioannis E., S.S.I.L.P.P.: Smoothing and stationarity enforcement framework for deep learning time-series forecasting. Neural Computing and Applications (2021)
10. Low, R.K.Y. Tan, E.: The role of analyst forecasts in the momentum effect. International Review of Financial Analysis **9** (2016)
11. Lucarelli, G., Borrotti, M.: A deep q-learning portfolio management framework for the cryptocurrency market. Neural Computing and Applications **32**(23), 17229–17244 (2020)
12. Mudassir M., Bennbaia S. Unal D., H.M.: Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. Neural Computing and Applications (2020)
13. Mulloy, P.: Technical Analysis of Stocks Commodities **40**(1) (1982)
14. Murphy, J.J.: Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. Penguin (1999)
15. Noel, A.D., van Hoof, C., Millidge, B.: Online reinforcement learning with sparse rewards through an active inference capsule (2021)
16. Sattarov, O., Muminov, A., Lee, C.W., Kang, H.K., Oh, R., Ahn, J., Oh, H.J., Jeon, H.S.: Recommending cryptocurrency trading points with deep reinforcement learning approach. Applied Sciences **10**(4), 1506 (2020)
17. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)