International Journal of Artificial Intelligence,
ISSN 0974-0635; Int. J. Artif. Intell.
2024, Volume 22, Issue Number 2

Forecasting Stochastic Time Series using Reinforcement Learning

N. Boyko[1] and **V. Kachmaryk**[1]

[1]Department of Artificial Intelligence Systems, Lviv Polytechnic National University,
79013, 12 Stepan Bandera Str., Lviv, Ukraine
Emails: nataliyaboyko3@ukr.net, viktorkachmaryk3@outlook.com

ABSTRACT

This study aims to analyse the effectiveness of machine learning algorithms, specifically deep reinforcement learning (DRL), in predicting financial time series. This research focuses on identifying the actions that must be taken at specific market states to maximize profit. Analysing these algorithms in the context of financial time series prediction is a crucial step toward improving investment strategies and maximizing returns. The outcomes of this research will provide insights into the potential of DRL for financial forecasting and inform the development of more accurate predictive models. In this work, authors applied deep reinforcement learning, specifically the Proximal Policy Optimization algorithm, to predict financial time series. The authors analysed the structure of the time series, performed data pre-processing, and developed a custom environment that mimics the logic of a stock exchange. The authors evaluated the performance of trained reinforcement learning agents using four basic models based on two strategies: buy and hold and random actions. The authors' agent outperformed the benchmark models significantly, and the authors assessed its risk by analysing the number of negative and positive returns for each day of testing. The results showed that the risk of its use is minimal but present. This study highlights the importance of a detailed analysis of the subject area, pre-processing of data, and development of a custom environment for financial time series prediction. Reinforcement learning shows promise in addressing the challenges of financial time series forecasting and should be further explored with larger data sets, multiple cryptocurrencies, and real-time deployment.

Keywords: Financial time series prediction, proximal policy optimization, long short-term memory, performance evaluation, risk assessment.

Mathematics Subject Classification: 37B05, 62M20.

Journal of Economic Literature (JEL) Classification: C02, C13

1. INTRODUCTION

In today's fast-paced world, relying solely on savings for financial security is increasingly difficult (Alieksieiev and Mazur, 2022). The present article explores the need to invest as an alternative to saving for the following two reasons: firstly, money sitting idle in a bank account means lost opportunities for earning more money. And secondly, savings do not have the potential to beat inflation, as shown by the inflation index (consumer price index)), which demonstrates an inflation rate of 20.5% over the last decade (Federal Reserve Bank…, 2023). This indicates the necessity to change the approach toward saving. Investing and trading are two distinct methods of profit-making in financial markets (Hayes,

2023). While investors and traders aim to generate returns through market participation, they differ in their approaches (Loi, 2023). Typically, investors pursue higher returns over the long term by purchasing and holding financial instruments. Conversely, traders capitalize on rising and falling markets by entering and exiting positions within a shorter timeframe, resulting in smaller but more frequent profits. In financial markets, two distinct approaches exist for earning profits – investing and trading. While both strategies aim to make money through market participation, they differ significantly in their approach – investors seeking to generate higher returns over a longer time horizon by holding financial instruments. At the same time, traders capitalize on upward and downward movements in markets to open and close positions more frequently, thus generating smaller but more frequent profits. Trading typically involves buying and selling securities, commodities, currency pairs, or other financial instruments to outperform the returns generated by a "buy and hold" investment strategy (Beniwal et al., 2023). While investors may be satisfied with annual returns of 10% to 15%, traders aim to earn 5% or more monthly (Chen, 2020). Profit is generated by purchasing any of the aforementioned financial instruments at a lower price and selling them at a higher price within a relatively short period (Kurhan et al., 2023). Alternatively, trading profits may be generated by selling any of the instruments mentioned above at a higher price and buying them back at a lower price (a strategy known as "selling short") to earn profits when the market declines (Artomov, 2022).

Financial time series represents a common type of non-stationary stochastic process that exhibit significant volatility and lack a consistent average level over an extended period. Extensive research on non-linear time series analysis has established mathematical models to describe the dynamics of these processes, ultimately enabling the forecasting of future prices based on current and past values (Boyko and Kachmaryk, 2022). While some researchers remain sceptical of this approach, there is ample reliable evidence to support the rejection of the hypothesis that financial time series corresponds to a random walk. This evidence underscores the value of robust modeling techniques that incorporate current and historical data to predict future movements in financial markets, ultimately informing investment strategies and decision-making processes.

This study focuses on developing and implementing a reinforcement learning (RL) agent capable of monitoring historical price and volume movements of cryptocurrencies, specifically Bitcoin (BTC), and executing real-time actions based on these values. This research explores the fundamental principles of constructing an RL environment and agent for analysing time series data, including the underlying algorithms that support the development of these models. The objectives of the research are:

1. Conduct a comprehensive analysis of available machine learning methods by conducting an in-depth review of scientific sources, evaluating their strengths and weaknesses, and presenting a comparative analysis.
2. Investigate the potential of deep learning with reinforcement in the context of working with stochastic time series, and provide examples of its practical applications.
3. Evaluate the performance of deep learning with reinforcement in processing and forecasting financial time series, comparing it with other established methods.
4. Develop a pattern analysis approach to address the challenge of handling high-dimensional financial time series data.

2. LITERATURE REVIEW

The study by H. Li et al. (2007) presents an application that uses reinforcement learning (RL) for real-time financial market prediction. The study considers both traditional eligibility criteria and a goal-oriented learning approach. The effectiveness of the "actor-critic" algorithms is compared with other studied alternatives. The results suggest that the "actor-critic" algorithms are more effective than other studied alternatives, including "actor-only" algorithms. This highlights the importance of considering both the actor and critic components in RL-based financial market prediction applications. Additionally, using goal-oriented learning methods can provide a more effective solution than traditional eligibility criteria.

In the study by D. Hendricks and D. Wilcox (2014), the authors propose a method for volume distribution between market orders by extending a model to Q-learning. The study is based on one year of market data reduced to 5-minute slices with five levels of market limit orders. The results of this study indicate that the proposed approach can improve the base model's performance by 10.3%. This suggests that Q-learning can be useful in developing models for volume distribution in financial markets. However, it should be noted that this study is based on a specific dataset and further research is needed to validate the effectiveness of this approach in other markets and time periods.

The study by J. Schulman et al. (2017) introduces a new family of policy gradient methods for reinforcement learning, where data is sequentially sampled through interaction with the environment, and the surrogate objective function is optimized using stochastic gradient growth. The paper compares PPO with other online methods of gradient politics and evaluates the balance between sampling complexity, simplicity, and running time. This study's proposed policy gradient methods provide a promising reinforcement learning approach. Using sequential sampling and minibatch updates in the objective function optimization yields a more efficient learning process than traditional methods. Compared with other online methods of gradient politics, PPO performs favourably in terms of balance between complexity, simplicity, and running time.

The study by X.Y. Liu et al. (2018) explores the potential of deep reinforcement learning in optimizing stock trading strategies. The study aims to maximize investment returns in a complex and dynamic stock market. The proposed approach is based on deep reinforcement learning, and the study compares its performance against two baseline models. The results from the study show that the proposed approach based on deep reinforcement learning outperforms the two baseline models in terms of both the Sharpe ratio and profit. This indicates the potential of deep reinforcement learning in developing more effective stock trading strategies in complex and dynamic markets. The study provides insights into the use of deep reinforcement learning as a tool to improve investment returns and inform financial decision-making.

The study by J. Sadighian (2020) proposes a new approach to constructing a reinforcement learning (RL) environment for modeling the operation of the cryptocurrency market. This environment is event-based, where events are defined as price changes that exceed a specific threshold. The paper also discusses different reward functions that consider the current state of the environment and the agent's actions based on unrealized profits, goals, and risks. The proposed approach in the article of constructing an event-based RL environment for modeling the cryptocurrency market provides a unique

perspective. Using event-based modeling provides a more realistic and practical way to simulate the cryptocurrency market's operation, which can help researchers develop better trading strategies. The proposed reward functions provide different approaches to optimizing agent performance, depending on the agent's goals and risk tolerance.

The authors have analysed the current state, and the majority of previous research analysing reinforcement learning (RL) in financial time series has focused on identifying ways to apply RL algorithms in this domain. However, the present study goes beyond this objective and aims to build an RL agent that can provide investors with greater assurance of success. Specifically, this work entails developing an application that observes the historical movements of cryptocurrency prices and volumes and executes actions in real-time based on these observations to achieve a net profit.

3. MATERIALS AND METHODS

When dealing with time series, it is crucial to ensure they are not white noise or random walk since predicting such time series can be partially or entirely impossible. White noise refers to data variation that cannot be explained by any regression model, although there exists a statistical white noise model for time series (1):

$$Yi = Li + Ni, \tag{1}$$

where: Y_i – observed value per step I; L_i – the value of the current level; N_i – random component (random value).

Figure 1 displays an example of white noise.

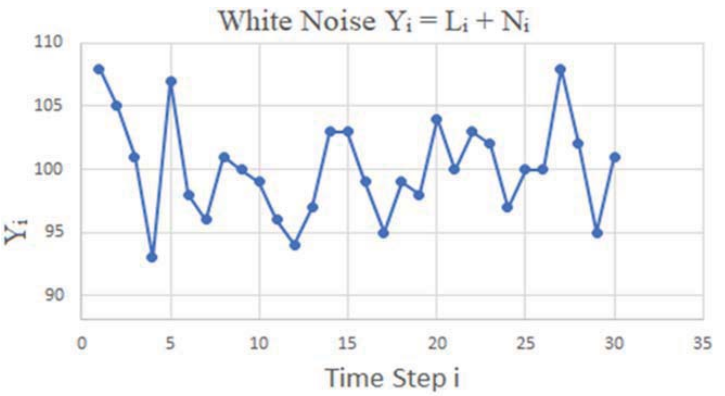

Figure 1. An example of white noise

Hence, it is crucial to note that white noise is a stationary time series with zero autocorrelation and constant variance. As illustrated in Figure 1, white noise exhibits randomness but without any clear pattern, unlike the other components of the time series. On the other hand, a random walk is a time series with stochastic behavior that describes a trajectory formed by successive random steps. The model for a random walk is described in (2):

$$Yi = Yi - 1 + Ni, \tag{2}$$

where: Y_i – the observed value in the previous step i-1.

Figure 2 shows an example of a random walk.

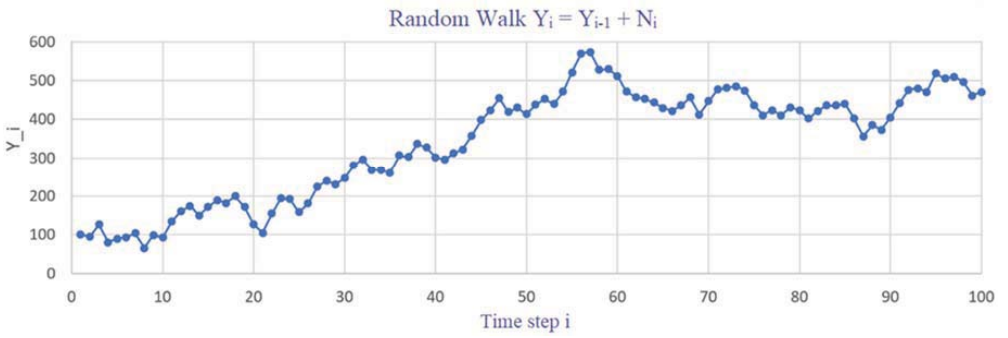

Figure 2. An example of a random walk

As previously mentioned, stochastic actions can be observed in a time series, such as the upward trend seen in the random walk example in Figure 2. Hence, performing preliminary data analysis to build accurate prediction models is crucial. Autocorrelation is one of the primary methods used in this analysis, along with other tests like Leung-Box and Box-Pearce. Autocorrelation is a measure of the similarity between a time series and its lagged version during subsequent time intervals. It is represented mathematically and is similar in concept to the correlation between two different time series but with the same time series used twice. On the other hand, partial autocorrelation describes the relationship between an observation in a time series and data from previous time steps without accounting for correlations between intermediate observations. Successive time step observations have a linear correlation function of indirect correlations. A detailed analysis of the input data will be conducted before creating the framework for training the agent and the RL environment. Figure 3 displays one of the datasets (one week of data) that will be used for training and testing the RL agent.

	open_time	open	high	low	close	volume	close_time	qav	num_trades	taker_base_vol	taker_quote_vol
0	2021-01-11 00:00:00+00:00	38150.02	38264.74	38143.48	38249.76	100.379301	2021-01-11 00:00:59.999000+00:00	3.835280e+06	1922	63.402755	2.422460e+06
1	2021-01-11 00:01:00+00:00	38249.76	38258.86	38052.65	38056.18	65.622697	2021-01-11 00:01:59.999000+00:00	2.502821e+06	1771	23.099152	8.807082e+05
2	2021-01-11 00:02:00+00:00	38056.02	38078.06	37869.81	38018.08	175.013816	2021-01-11 00:02:59.999000+00:00	6.645152e+06	3126	74.627570	2.833196e+06
3	2021-01-11 00:03:00+00:00	38018.08	38046.19	37884.50	37899.99	81.197138	2021-01-11 00:03:59.999000+00:00	3.082775e+06	1628	39.395046	1.495800e+06
4	2021-01-11 00:04:00+00:00	37899.99	37959.24	37700.00	37724.99	193.242820	2021-01-11 00:04:59.999000+00:00	7.303177e+06	3582	92.702169	3.503084e+06
...	...	...	...	...	...	...	...	...	...	...	...
10075	2021-01-17 23:55:00+00:00	35772.60	35810.58	35753.26	35800.67	37.711206	2021-01-17 23:55:59.999000+00:00	1.349417e+06	896	20.838865	7.456716e+05
10076	2021-01-17 23:56:00+00:00	35800.67	35800.67	35633.40	35712.23	59.278659	2021-01-17 23:56:59.999000+00:00	2.116583e+06	1514	26.265766	9.378545e+05
10077	2021-01-17 23:57:00+00:00	35715.98	35786.91	35675.36	35784.02	37.464863	2021-01-17 23:57:59.999000+00:00	1.338639e+06	1086	20.211238	7.222131e+05
10078	2021-01-17 23:58:00+00:00	35783.12	35832.71	35774.21	35814.43	51.212251	2021-01-17 23:58:59.999000+00:00	1.833838e+06	1226	27.969865	1.001544e+06
10079	2021-01-17 23:59:00+00:00	35813.47	35851.03	35810.00	35828.61	27.656391	2021-01-17 23:59:59.999000+00:00	9.909203e+05	778	12.017219	4.305528e+05

10080 rows × 11 columns

Figure 3. Dataset (one week of data).

For a comprehensive analysis of the input data, autocorrelation and partial autocorrelation graphs were generated for all the features in the dataset (Figure 4).

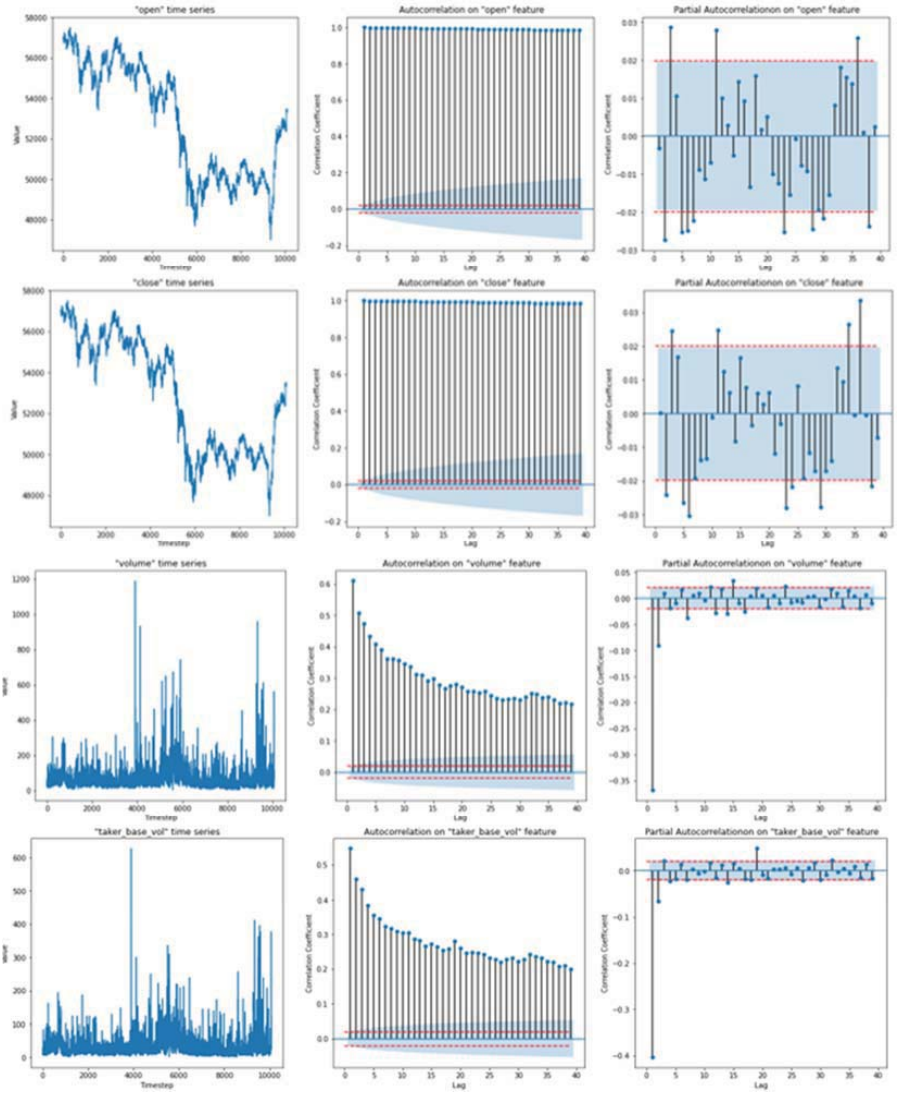

Figure 4. Visualization of input dataset properties with their corresponding autocorrelation and partial autocorrelation plots.

Figure 4 provides a visualization of the autocorrelation and partial autocorrelation plots for all features of the input dataset, revealing important insights into their properties. Notably, the prices (open and close) correlate highly at various lags, indicating a possible association with a random walk. However, the corresponding partial autocorrelation plot resembles white noise with no significant correlation, which is not applicable in this case. Furthermore, it is essential to note that the distribution of the various properties in the dataset varies greatly. For instance, the values of properties like prices (open, high,

low close) range between 47000 and 58000, whereas the 'taker_base_vol' property ranges from 0 to 600. Therefore, due to this uneven distribution, normalizing all properties is necessary. The authors decided to use QuantileTransformer for this purpose, as it not only normalizes the properties to a distribution between 0 and 1 but also maintains their relative distribution while removing upper and lower outliers, as illustrated in Figure 5.

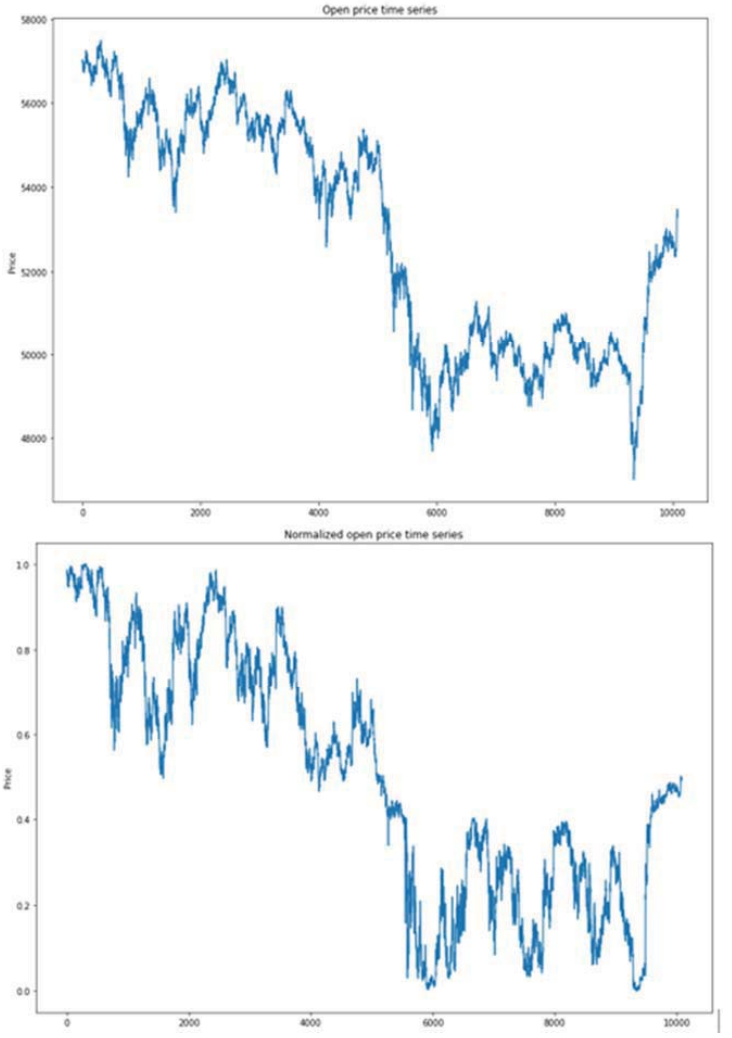

Figure 5. Graph "before" and "after" normalization of the "open" property.

To build an agent that will predict financial time series, it was decided to use a deep reinforcement learning model – Proximal Policy Optimization. PPO is a gradient policy method for reinforcement learning. The motivation behind its creation was to have an algorithm with the data efficiency and robust performance of the Trust Region Policy Optimization (TRPO) method using only first-order optimization. That is, this algorithm provides an improvement in the optimization of the TRPO policy. However, to

understand and appreciate this method, authors first need to understand the reinforcement learning policy.

4. RESULTS

3.1. Software implementation

The software implementation of the information system (IS) for predicting financial time series using deep learning with reinforcement was divided into four parts to accomplish the following tasks:

1. Establishment of a data-downloading pipeline to obtain the required data.
2. Development of a reinforcement learning (RL) environment to train an RL agent for predicting future prices of the designated cryptocurrency.
3. Development of a training pipeline to instruct the RL agent on the correct policy of actions and evaluation of its performance on new data.
4. Implementation of baseline models for comparison of results.

The study uses historical data on Bitcoin, the most popular cryptocurrency, for several reasons. Firstly, the total value of Bitcoin is approximately 1 trillion dollars, accounting for 50% of the value of all other cryptocurrencies. This high market capitalization has resulted in the value of other cryptocurrencies being partially dependent on Bitcoin, making it difficult to predict their value. Secondly, a financial time series' stochasticity is an essential factor affecting its prediction. Bitcoin has higher liquidity than other cryptocurrencies, making it more difficult to artificially change its value, reducing the time series' stochasticity. The historical data of Bitcoin cryptocurrency was downloaded using the python-Binance library, which utilizes the Binance REST API for accessing the historical data of all cryptocurrencies traded on the Binance trading exchange. This library allows downloading historical data with varying time intervals between events such as 1 min, 3 min, 5 min.

The downloaded historical data was structured using the "pandas.DataFrame" data structure, with each column representing one of the properties of the historical data, such as the opening and closing price of the event, amount of cryptocurrency. The data was then divided into weeks and saved as parquet files for quick usage in the RL environment. This approach allows for easy access and manipulation of the data in the RL environment. Before training the RL agent, an RL environment that simulates the operation of a cryptocurrency exchange must be created. However, it is essential to comprehend what an RL environment is. In reinforcement learning, the environment refers to the world of the agent with which it interacts. The agent can perform an action on the environment but cannot influence its rules or dynamics. When an agent takes action on an environment, it produces a new environment state, which causes the agent to transition to that state. The environment also sends the agent a reward, which is a scalar value that serves as feedback to the agent about the quality of its action. To better understand the agent's interaction with the environment, refer to the interaction diagram depicted in Figure 6.

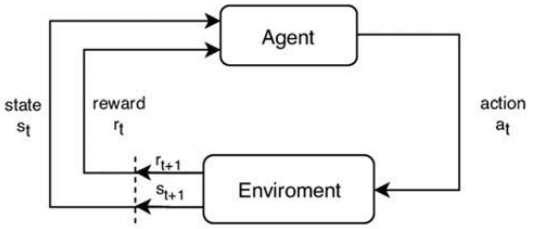

Figure 6. Scheme of the interaction of the agent with the environment

Hence, it can be asserted that the environment plays a crucial role in the application of reinforcement learning. The efficacy with which an agent can learn specific policies largely depends on the implementation of the reward function, which is an integral component of the environment. The Gym library was utilized to construct the RL environment, which enables the creation of a specific environment that encompasses all the essential functionalities for initializing and training the agent (Gym Documentation, 2023). Each environment must comply with the gym interface, which consists of four primary functions, namely:

1. The __init__ function is the constructor, where the type and structure of the action space (action space) are defined, encompassing all the actions that the agent can perform in the environment. Similarly, the observation space (observation space) is defined, which comprises all the environment data that the agent can observe at each step in the environment.
2. The reset function periodically resets the environment to its initial state.
3. The step function – where the agent's action based on the observation is executed and returns the following observation. A reward is also calculated at each step according to the agent's action and observation.
4. The render function is essential for rendering the state of the environment at the current step, allowing an understanding of the environment and the agent's state at each step. This function can be as simple as a print operator or as complex as rendering a 3D environment using OpenGL.

In the following stage, the type and structure of the action space, the structure of the observation space, what data this space will consist of, and the reward calculation function need to be defined. The action space represents a set of actions that an agent can perform in a given environment. It is specific to each environment. There are three types of action space:

- discrete action space corresponds to actions of a discrete nature;
- continuous action space corresponds to actions of a continuous nature;
- mixed action space consists of both discrete and continuous actions.

Upon careful analysis of the types of action space, the authors chose a discrete action space for this task. This space corresponds to the possible actions of the agent when working with the cryptocurrency exchange and consists of three possible actions: buy, sell, and hold (Herzen et al., 2022). The observation space is a critical component of the RL environment, representing the information available to the agent at each step. While the action space is a relatively deterministic part of the environment (consisting of only three types), the observation space is what the agent can observe, forming the world

in which the agent operates. For example, this can take the form of an Atari game or a self-driving car. To address the problem at hand, the observation space must contain all relevant input variables that the agent will use to determine its actions. For this study, historical data on the popular cryptocurrency Bitcoin was chosen as the basis for constructing the observation space. To provide the agent with a more comprehensive view of the environment, it is important to include not only the current properties of the cryptocurrency, such as price and volume but also past values. A sliding window approach will be used to create observation vectors to achieve this (Wagena et al., 2020).

Additionally, analysing the scientific sources, it was investigated that including metadata can improve the agent's training. Therefore, metadata will be included as additional values for the observation vectors, allowing for an evaluation of the impact of metadata on the agent's learning in the context of this problem. In this study, the authors aim to improve the agent's learning performance by utilizing metadata in addition to historical cryptocurrency data. To this end, authors have chosen to use the following metadata:

1. The day of the week is represented as a one-hot vector.
2. The number of steps remaining until the end of the episode, which is normalized by dividing by the maximum possible number of steps in the episode (the number of vectors in the observation space).
3. The agent's position, which can be either Long (when the agent has bought cryptocurrency and has an open position) or Neutral (when the agent does not have an open position), is represented as a one-hot vector.
4. Total cost, which is normalized by dividing by the maximum possible cost.

The idea behind incorporating metadata is to ensure that, at every time step, the agent considers not only the alterations in the cryptocurrency features but also the state of its portfolio, specifically, its position and total value. This information will enable the agent to make a well-informed decision about the following action. It should be emphasized that the agent's learning performance is contingent on constructing the reward function. Consequently, this study proposes two distinct reward functions. The first proposed reward function is based on the investor's interactions with the cryptocurrency exchange and consists of three conditions:

1. If the agent opens a position by buying cryptocurrency, the agent receives a reward of 0. While it may be more logical to assign a negative reward to the transaction (due to commission fees when opening or closing a position), such a reward scheme may discourage the agent from opening a position.
2. If the agent closes a position by selling the purchased cryptocurrency (assuming the agent has made a purchase), the agent receives a reward equal to the difference between the selling and purchase prices.
3. If the agent holds the position or performs no action (e.g. if there is no purchased cryptocurrency), the agent receives a negative reward or fine for holding.

The second reward function is more intricate than the first, as it involves a hidden algorithm to calculate the reward and uses both the earned profit and a benchmark profit for comparison. To implement this function, authors calculate the profit earned by the agent at each step and a reference profit computed using the buy-and-hold strategy (i.e., buying the cryptocurrency in the first step and selling it in the

current step). The authors then compute the difference between the earned profit and the reference profit, which serves as the reward for the agent. With this reward function, authors incentivize the agent to make optimal decisions at each step rather than simply earning a profit. In this study, the authors utilized historical data on Bitcoin, the most popular cryptocurrency, to train and evaluate the performance of the authors' agent. The data was obtained from [3.1.1] and stored in separate weeks. The authors' objective is to create a structure that enables the agent to predict the future price of the cryptocurrency and make a profit in the market. It is important to note that the price movement of the cryptocurrency varies across different weeks. Thus, it is necessary to examine the data from a comparative perspective. Figure 7 illustrates the change in the opening price across different weeks.

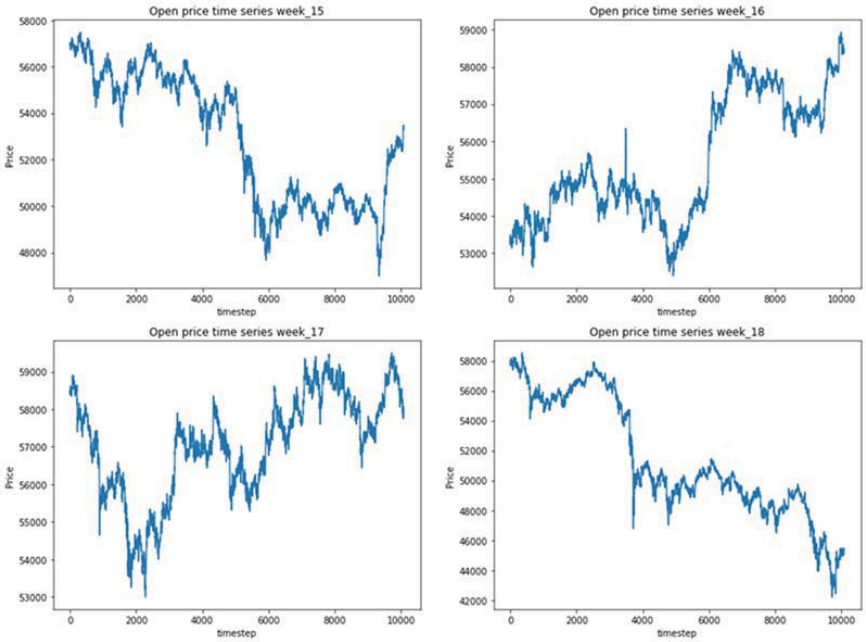

Figure 7. "Open" price changes on different weeks.

Figure 7 illustrates the change in the opening price of Bitcoin for other weeks. As can be seen, there is a negative trend in the price time series in the 15th and 18th weeks, while in the 16th week, there is a positive trend, and in the 17th week, the trend is volatile. This poses a challenge when training an agent to work with financial data since training on one week's data may not generalize well to other weeks. Another challenge is the episode length for RL training. In this case, the episode is one week long, consisting of only 10080 (60*24*7) events with a 1-minute interval distribution. This length is insufficient for the agent to learn a policy effectively. The authors propose using a random starting point during environment initialization and resetting to address this issue. This approach enables iterative training on the same week with different starting points, resulting in a different set of actions the agent needs to make to achieve profit. This technique can also overcome the lack of trend difference in price time series between individual weeks, as the agent will not become accustomed to any linear policy.

Therefore, authors can train an agent capable of predicting the future price of cryptocurrency and making a profit from its actions in the cryptocurrency market.

It should be noted that in order to normalize each property (feature) of the input data set, the Quantile Transformer was chosen. During the training process, the scaler was fit on the same dataset that was used to train the agent. For inference purposes, the Quantile Transformer that was trained on the training data will be used to normalize the input data. This approach helps prevent the model from having access to information about the future and ensures that the structure of the trained agent corresponds to the structure of the working application. To build an agent using the PPO algorithm, the open-source library Stable Baselines3, which provides implementations of various reinforcement learning algorithms, was utilized. It was decided that the training would take the form of an iterative review of available weeks. The agent is trained on the corresponding week at each step, and its performance is evaluated in the following five weeks. The agent's state is also saved after the training process to assess the agent's performance and potential future use.

4.2. Implementation of baseline models for comparison of results

Given the limitations of assessing an agent's performance based solely on its testing (inference) profits, authors opted to construct baseline models to compare their returns against those of the trained agent. For this purpose, the authors utilized buy-and-hold – one of the standard investment strategies and random actions to simulate a novice investor's actions. These baselines provide reference points to evaluate the performance of the trained agent. The buy-and-hold strategy is a well-known investment strategy in which an investor purchases a financial instrument and holds it for a specified period before selling it. To implement this strategy in the context of this study, the authors adopted two approaches: purchasing cryptocurrency at the beginning of each day and selling it at the end of the same day; purchasing cryptocurrency at the beginning of each week and selling it at the end of the same week. To implement a strategy based on random actions, the authors decided to utilize two approaches. The first approach involves maintaining a uniform probability distribution for the three possible actions (buying, holding, and selling) at each step in the environment. The second approach involves a non-uniform probability distribution for the three possible actions: [0.1, 0.8, 0.1] for buy, hold, and sell, respectively. This distribution was chosen because frequent buying and selling would increase transaction fees, making it less desirable. Applying these basic models will allow to show how well the trained agent copes with predicting the future price by comparing its profit with the profit of the basic models.

4.3. Experimental part

To perform a comprehensive analysis of the RL agent's ability to predict financial time series and make profitable trades, a series of experiments were conducted with varying hyperparameters for both the agent and the environment. To more thoroughly evaluate the agent's performance, baseline models were constructed and used as a point of reference for comparison. Two strategies were utilized to

construct these basic models, as outlined in section 3.2: “buy and hold” and strategy based on random actions. Based on these strategies, four basic models were developed:

1. Buy and hold (buy and hold) on a daily time interval (buying cryptocurrency at the beginning of the day and its subsequent sale at the end of the day).
2. Buy and hold (buy and hold) on a weekly time frame (buying cryptocurrency at the beginning of the week and its subsequent sale at the end of the week).
3. Uniform distribution of random actions (at each step in the environment, the probability distribution for three possible actions (buying, holding, and selling) is uniform).
4. Uneven distribution of random actions (at each step in the environment, the probability distribution for the three possible actions is not uniform and is equal to [0.1, 0.8, 0.1] for buying, holding, and selling, respectively).

Throughout this paper, all financial values are reported in US dollars unless otherwise stated. Authors will refer to these values simply as “conditional units” for clarity. A dataset of five weeks of historical Bitcoin cryptocurrency data was used to assess the performance of both the base models and the RL agent. This data was not previously seen or used to train the RL agent. At the start of each week, the initial portfolio value was set at 70000 conditional units. The results of the base models and RL agent performance will now be examined in detail. This methodology aims to acquire a financial asset, specifically cryptocurrency, at the onset of a given day and divest it at the day’s end. By following this technique, authors have generated the outcome depicted in Figure 8.

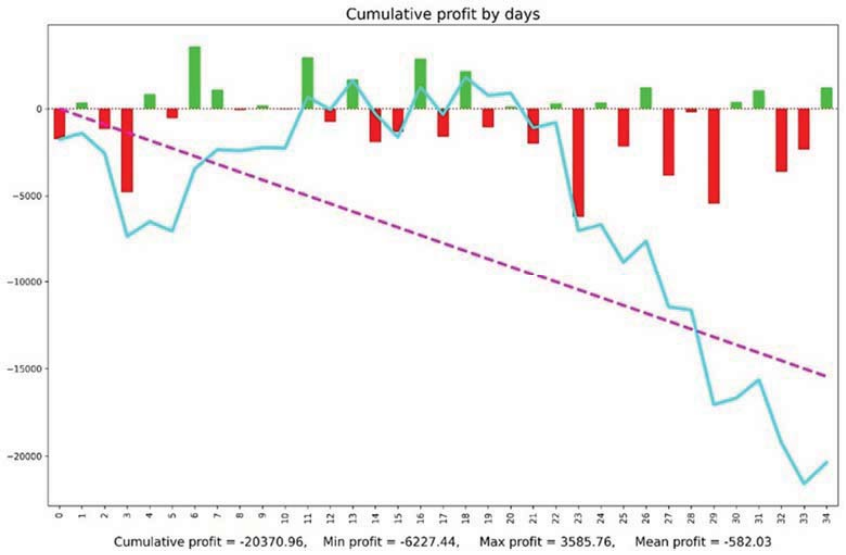

Figure 8. Cumulative income of an agent for five weeks.

Figure 8 shows the daily profits of the studied model over a testing period of 5 weeks, where positive profits are depicted in green and negative profits in red. As evident from the graph, the cumulative profit of this approach, which involves buying cryptocurrency at the beginning of the day and selling it at the

end of the day, is equal to (-20371) conditional units, suggesting its unsuitability for trading during the investigated period. Although the cumulative profit initially grows and surpasses the initial investment, it plummets steeply towards the end of the period, attributable to a downward trend observed in the last three weeks of testing after an initial upward trend during the first two weeks. Moreover, the approach yields a maximum profit of 3586, a minimum profit of (-6227) conventional units, and an average profit of (-582) conventional units. Further details of the agent's actions constructed based on the "buy and hold on a daily time frame" approach during one week of data are presented in Figure 9.

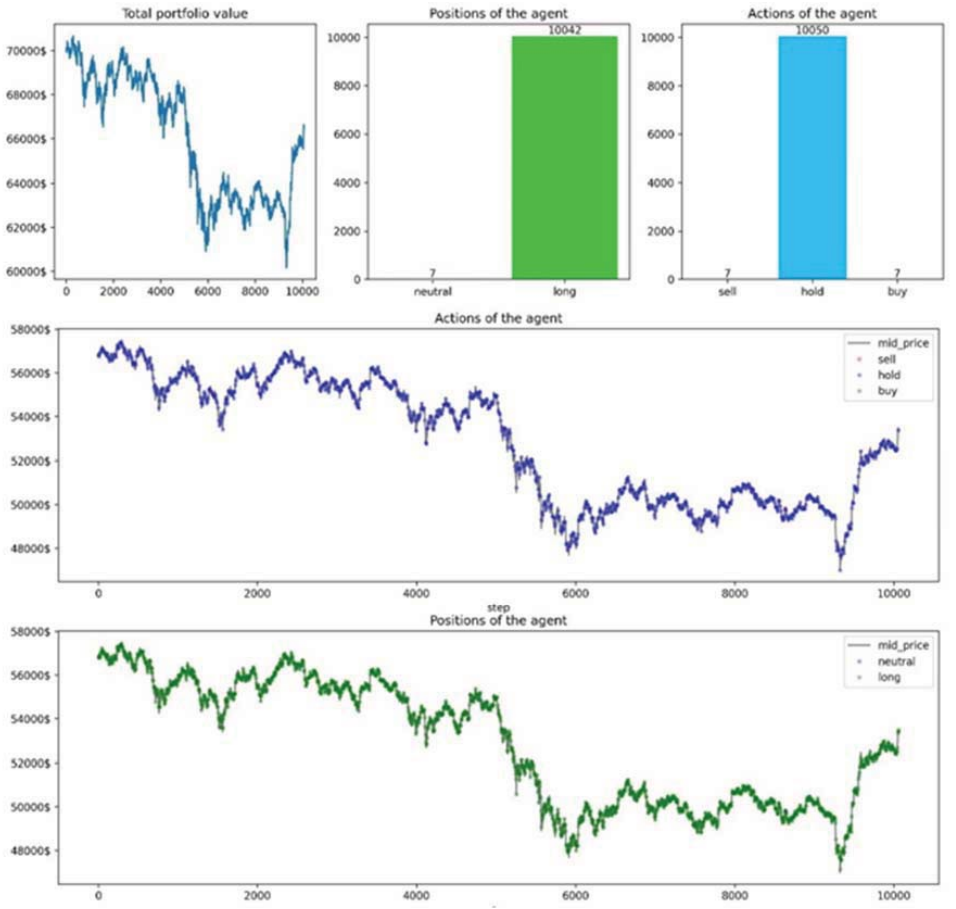

Figure 9. Detailed analysis of agent performance on one week of data.

Figure 9 illustrates the agent's actions following the "buy and hold on a daily time frame" approach. This approach involves buying cryptocurrency at the beginning of the day and selling it at the end of the day. The figure shows the changes in the portfolio value as well as the actions taken by the agent concerning the price of cryptocurrency and the positions held. This methodology aims to acquire a financial asset, specifically cryptocurrency, at the onset of a given week and divest it at the week's end. Figure 10

presents the outcome of applying this approach. Positive profits are displayed in green, while negative profits are displayed in red.

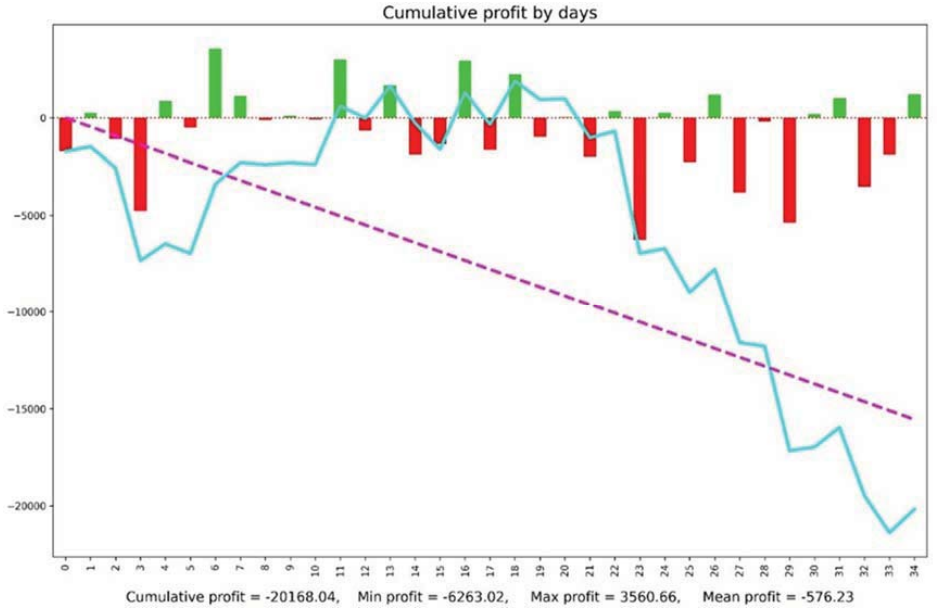

Figure 10. Cumulative income of an agent for five weeks.

As shown in Figure 10, the overall trend of the cumulative profit of the agent constructed based on the "buy and hold on a weekly basis" approach is quite similar to that of the agent built using the "buy and hold on a daily basis" approach. Initially, the cumulative profit increases and exceeds the portfolio's initial value, but it sharply declines at the end. The following results are obtained when evaluating the performance of this agent:

1. The cumulative profit is (-20168) conditional units.
2. The minimum profit is (-6263) conditional units.
3. The maximum profit is 3561 conditional units.
4. The average profit is (-576) conventional units.

In Figure 11, the authors can examine the detailed actions of the agent based on the "buy and hold on a weekly time frame" approach over one week of data.

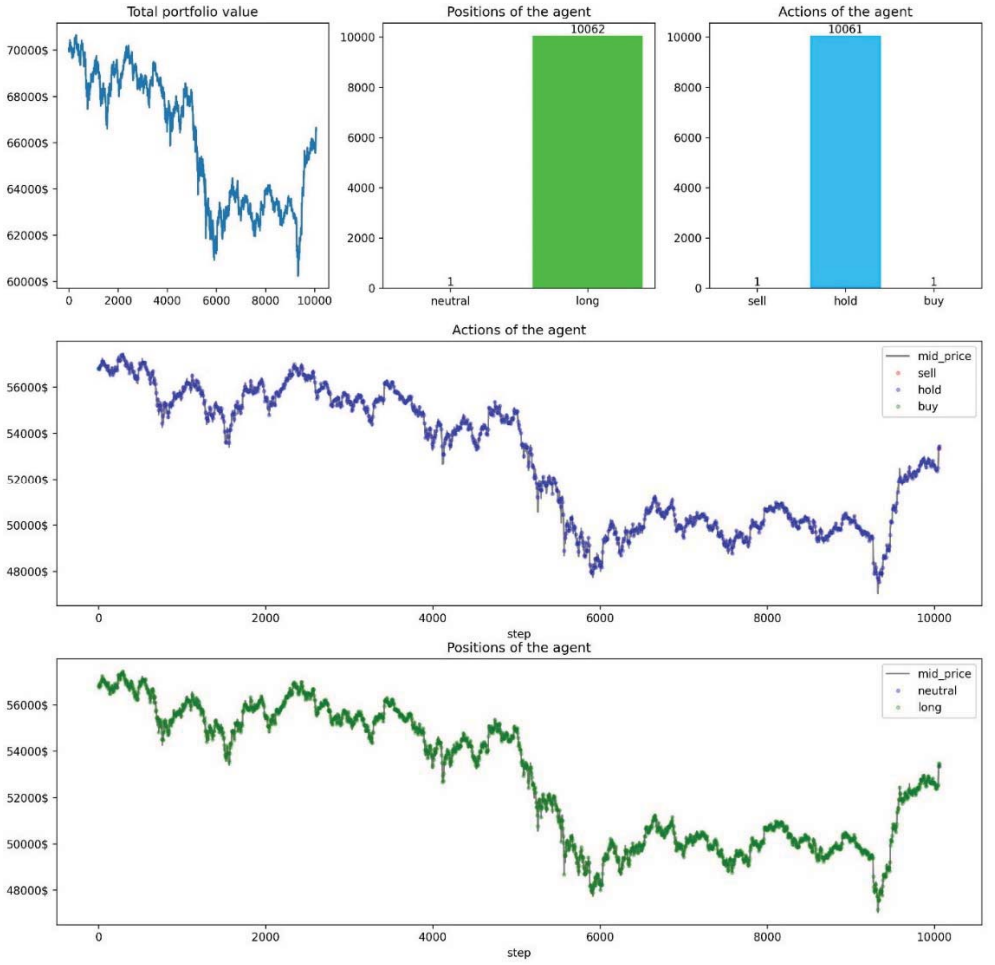

Figure 11. Detailed analysis of agent performance on one week of data.

Figure 11 illustrates the agent's actions, which align with the approach employed for constructing the agent, namely buying cryptocurrency at the beginning of the week and selling it at the end of the week. This figure depicts how the agent's actions influenced the value of the portfolio and the agent's actions regarding the cryptocurrency price and positions held. The objective of this approach is for the agent to randomly select one of the three possible actions (buy, hold, and sell) at each time step of the environment. By implementing this approach, the authors acquired the results presented in Figure 12.

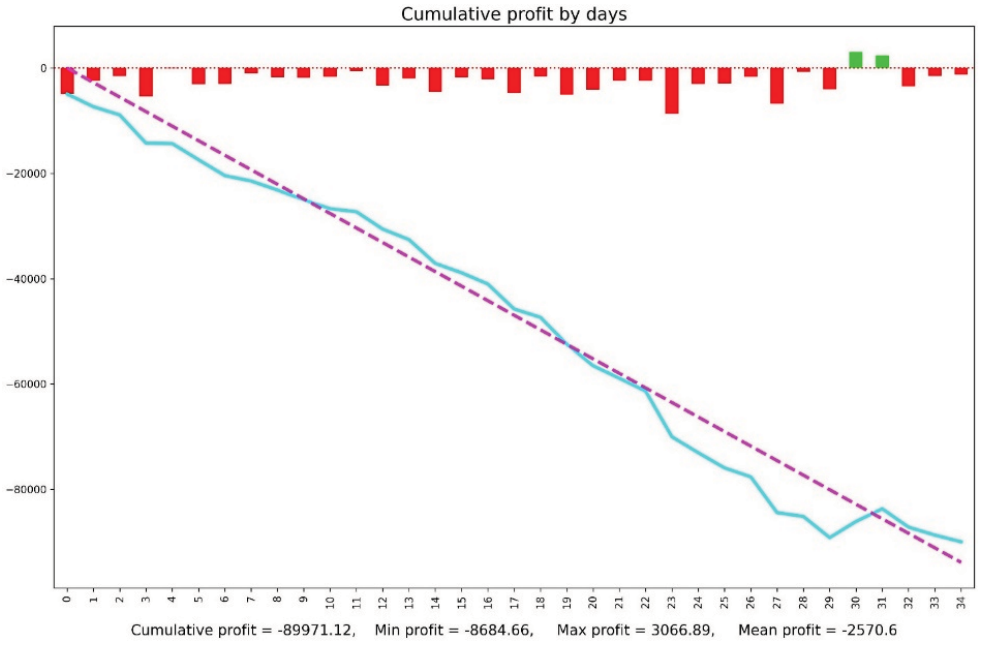

Figure 12. Cumulative income of an agent for five weeks.

In Figure 12, it is evident that the cumulative profit of the agent, built based on the "uniform distribution of random actions" approach, is declining, indicating that this approach is unsuitable for analysing such time series. One of the main reasons for this is the presence of exchange fees that are calculated as a percentage; therefore, frequent cryptocurrency purchases and sales are constrained. The summary of the agent's performance is as follows:

1. Cumulative profit is (-89971) conditional units
2. The minimum profit is (-8684) conditional units.
3. The maximum profit is 3067 conditional units.
4. The average profit is (-2571) conventional units.

A detailed analysis of the agent's actions based on the "uniform distribution of random actions" approach for a week of data is presented in Figure 13.

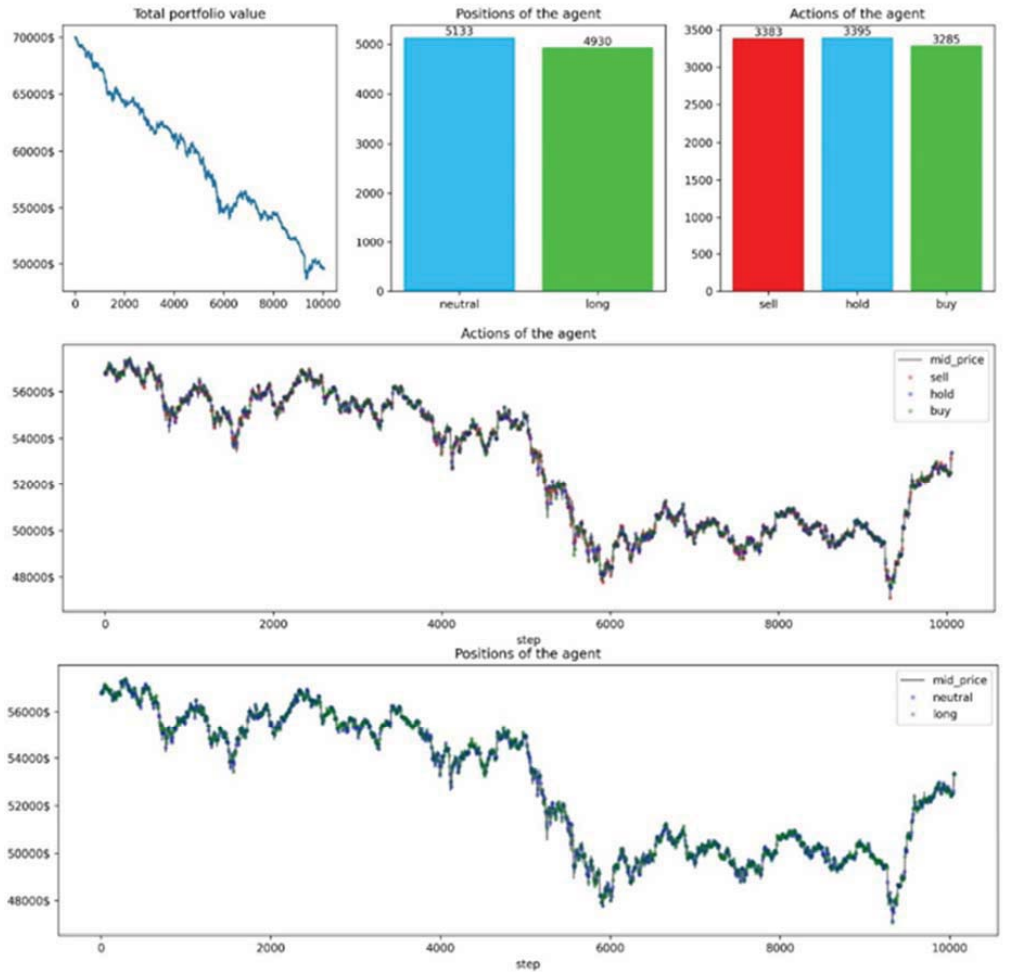

Figure 13. Detailed analysis of agent performance on one week of data.

Figure 13 displays the agent's actions in accordance with the approach used in constructing the agent, the changes in the portfolio value, and the agent's actions in relation to the cryptocurrency price and its positions. The objective of this approach is for the agent to take one of the three possible actions (buy, hold, and sell) randomly at each step of the environment, but the probability of "buying" or "selling" is much lower than the probability of holding ([0.1, 0.8, 0.1] – distribution for buying, holding, and selling, respectively). This distribution is justified by the commission, which authors must pay with each transaction, making buying and frequently selling unsuitable. The outcome of this approach is presented in Figure 14.

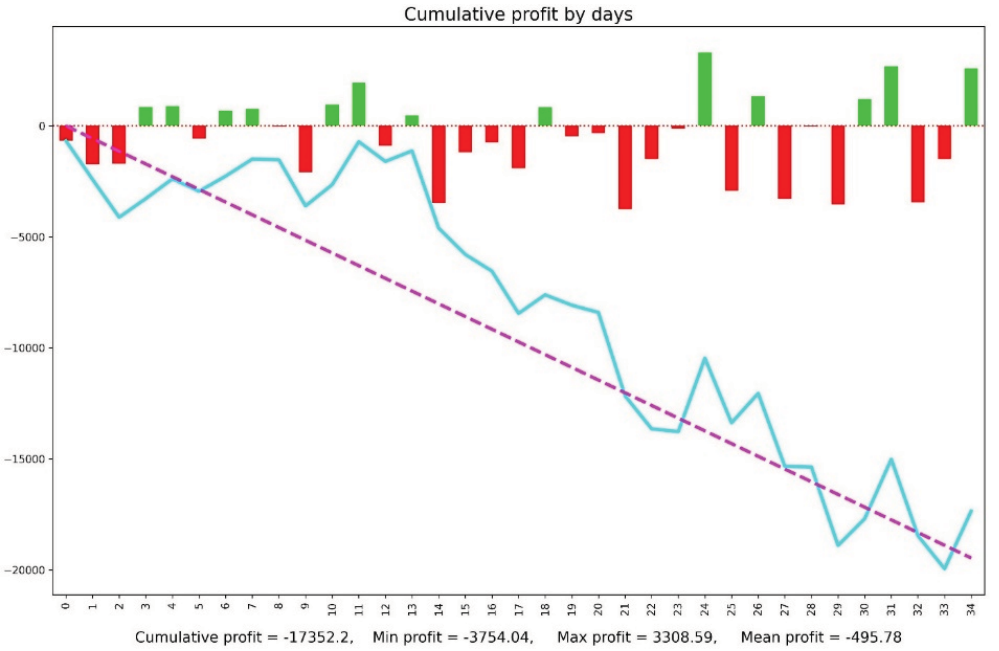

Figure 14. Cumulative income of an agent for five weeks.

Figure 14 shows that the approach with an uneven distribution of random actions is much more effective than the one with a uniform distribution. Although the cumulative profit trend of the agent is downward, there are many days on which the profit is positive. This is because the agent makes significantly fewer transactions (buying and selling), which means that it incurs lower transaction fees. Additionally, with fewer transactions, there is a higher probability of obtaining positive profits. The summary of this agent's performance is presented below:

1. Cumulative profit equals (-17352) conditional units.
2. The minimum profit is (-3754) conditional units.
3. The maximum profit is 3309 conditional units.
4. The average profit is (-496) conditional units.

Detailed actions of the agent based on the "uneven distribution of random actions" approach in one week of data are presented in Figure 15.

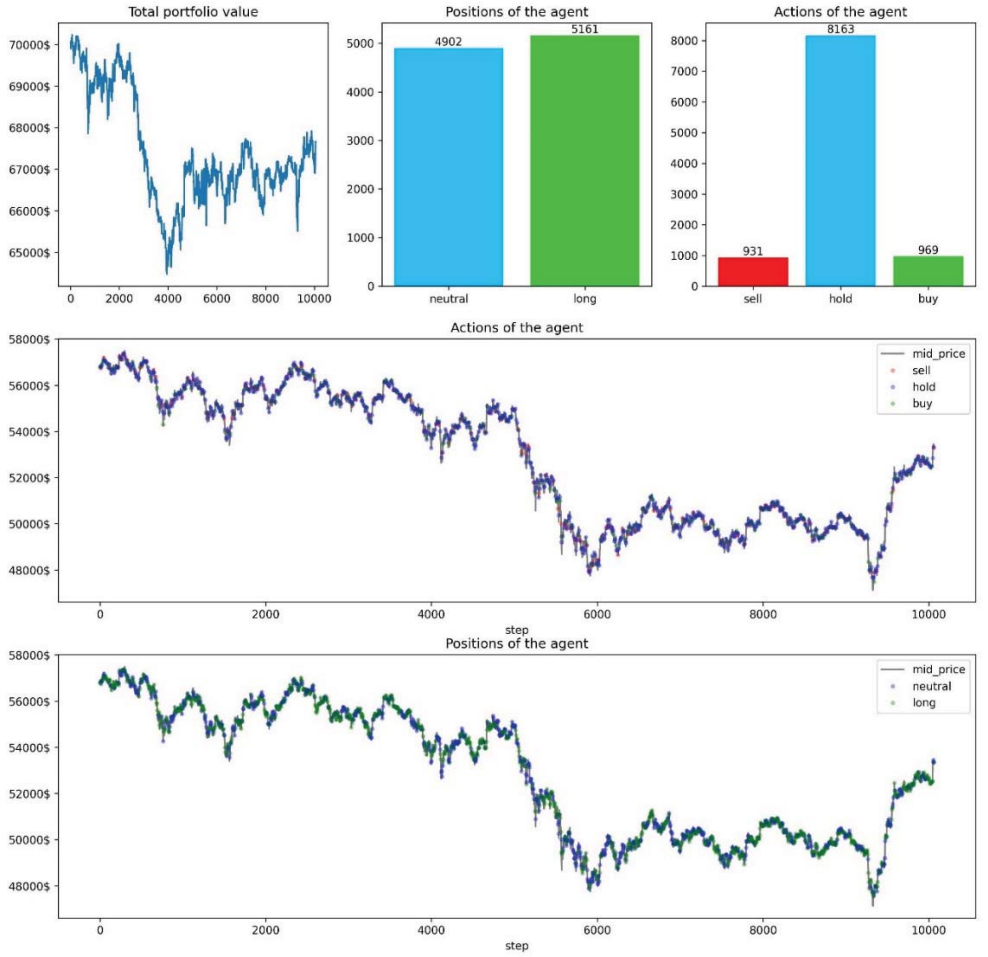

Figure 15. Detailed analysis of agent performance on one week of data.

Figure 15 depicts the actions taken by the agent, corresponding to the approach utilized in its construction, as well as the changes in portfolio value and the agent's actions relative to the cryptocurrency price and positions. The RL agent was trained iteratively by searching through the available weeks. The agent was trained on the corresponding week at each step, and its performance was evaluated within the next five weeks. This approach allowed the agent to receive new data that it had not previously encountered during training. To better understand the agent's training process, authors used TensorBoard to monitor essential values such as "training loss", "mean episode reward", and "value loss". The results are presented in Figures 16-18.

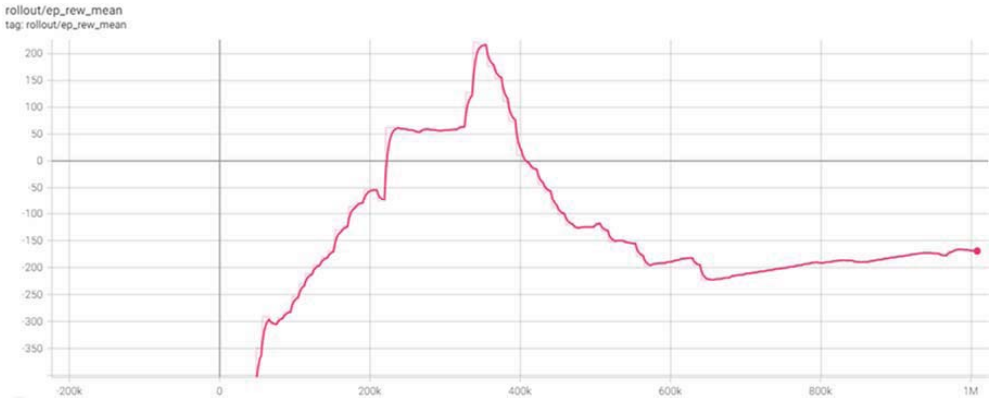

Figure 16. The average reward value per episode.

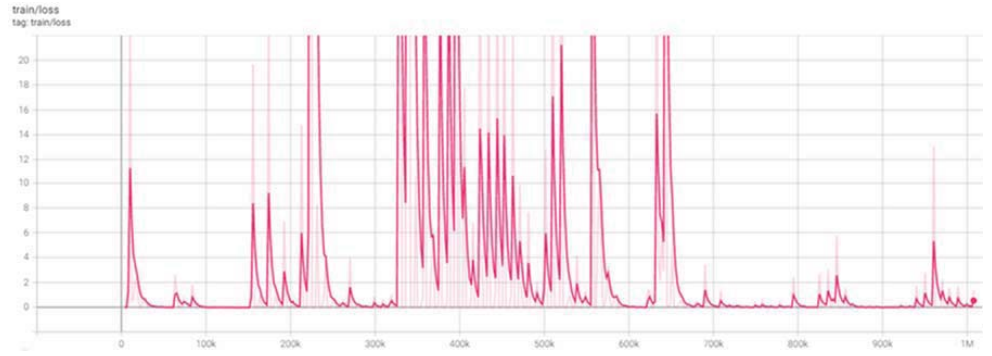

Figure 17. The loss function of policy gradients (training loss).

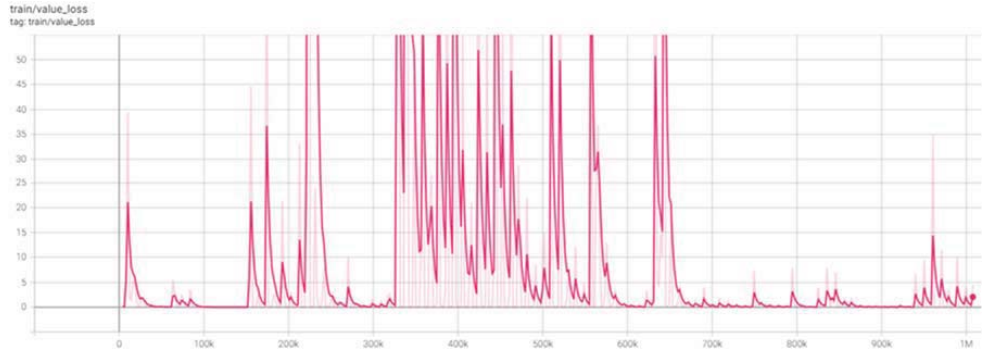

Figure 18. Average losses (loss) of the value update function.

Figures 17 and 18 show that the agent was trained correctly, as evidenced by the convergence of the loss functions before the end of training. In this section, the authors will present the results of the RL

agent's performance during the 5-week period when no further training was conducted. These results are presented in Figure 19.

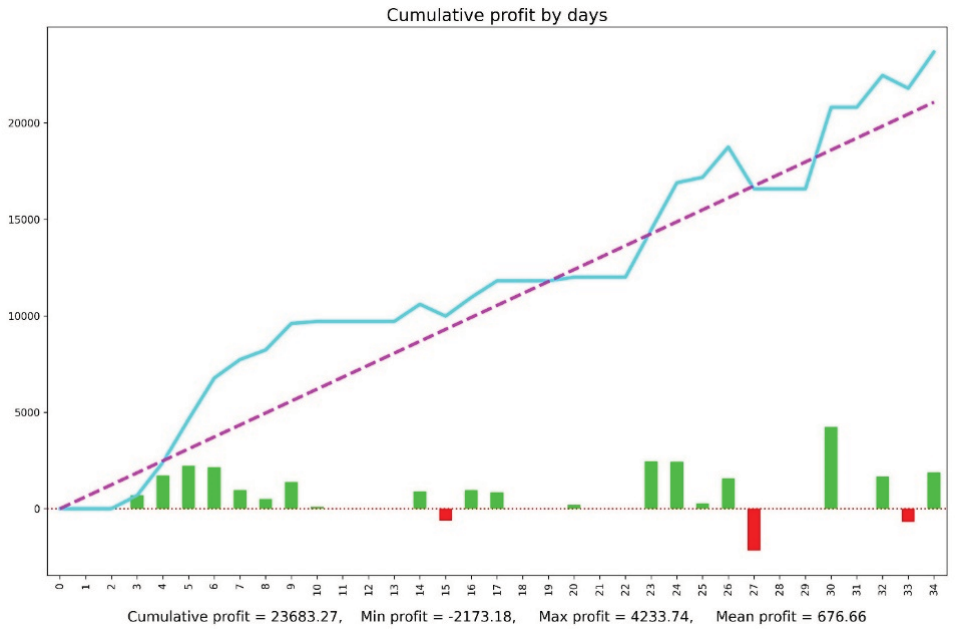

Figure 19. Cumulative income of an agent for five weeks.

In Figure 19, the authors observe that the trained RL agent performs significantly better than all previous base models, as evidenced by the upward trend of the aggregate profit. Additionally, there are only three days with negative profit and 19 with positive profit, which indicates that the agent has found a suitable policy that allows for profit on most days, thus minimizing risks during operation. In summary, the results of this agent's work are as follows:

1. The cumulative profit is 23683 conditional units.
2. The minimum profit is (-2173) conditional units.
3. Maximum profit is 4234 conditional units.
4. The average profit is 677 conditional units.

Figure 20 provides a detailed overview of the RL agent's actions during one data week.

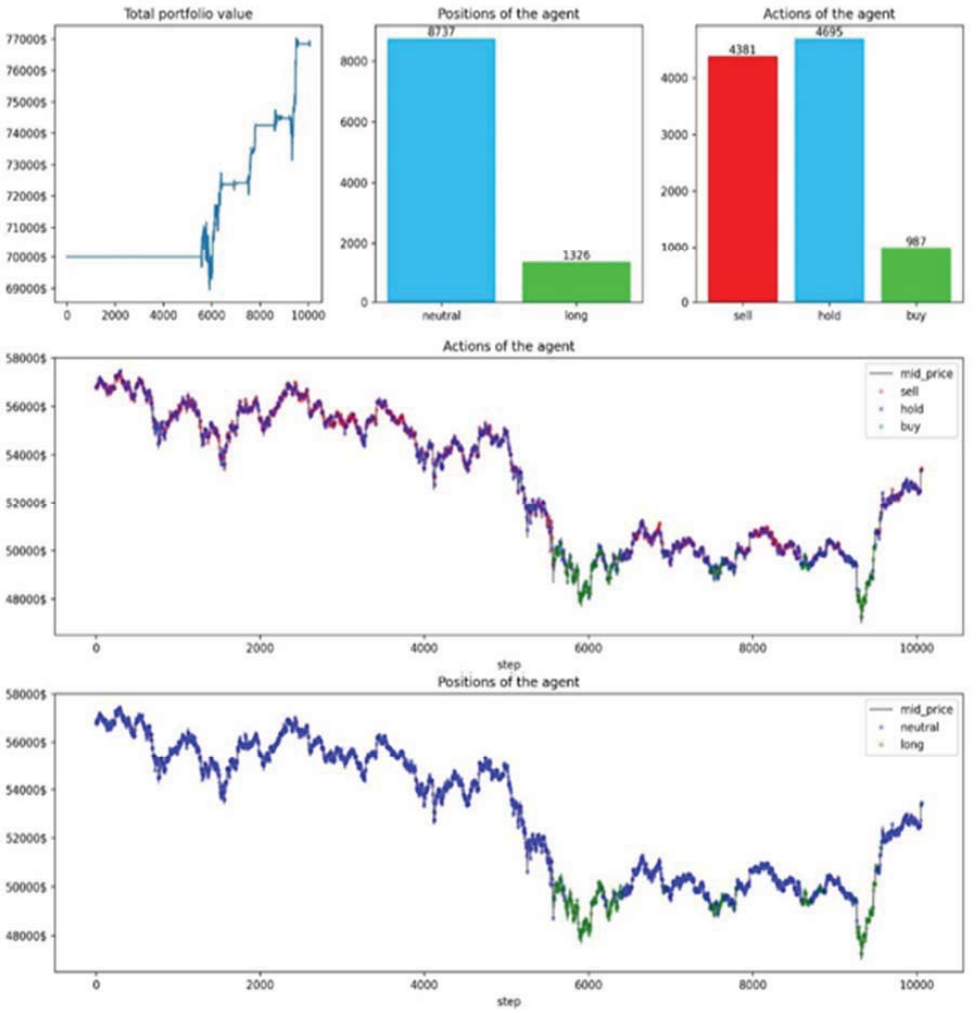

Figure 20. Detailed analysis of agent performance on one week of data.

Figure 20 presents the actions taken by the RL agent, the changes in the portfolio value, and the agent's actions concerning the cryptocurrency price and positions held. It is evident that the RL agent performs intelligent actions, such as buying the cryptocurrency before its price rises and selling it before its price falls. This indicates the success of creating an RL agent that can predict financial time series and take actions in the financial market to generate a net profit.

4.3. Summary of results

Table 1 presents the performance results of the benchmark base models and the trained RL agent's performance results during the testing phase over five weeks. Additionally, the RL agent incurred the lowest losses with negative returns, indicating minimal risk during its operation. Table 2 displays the base models' and RL agents' performance.

Table 1: Results of the basic models and the trained RL agent.

Agent's model	Cumulative profit	Maximum profit	Minimum profit	Average profit
Buy and Hold (daily basis)	-20371	3586	-6227	-582
Buy and Hold (weekly basis)	-20168	3561	-6263	-576
Uniform distribution of random actions	-89971	3067	-8685	-2571
Non-uniform distribution of random actions	-17352	3309	-3754	-496
RL agent	23683	4234	-2173	677

Table 2: Performance of basic models and RL agent.

	Buy and Hold (daily basis)	Buy and Hold (weekly basis)	Uniform distribution of random actions	Non-uniform distribution of random actions	RL agent
The initial value of the portfolio	70000	70000	70000	70000	70000
The final value of the portfolio	49629	49832	-19971	52648	93683
Profit	-29.1%	-28.8%	-128.5%	-24.8%	33.8%

Based on the results presented in Tables 1 and 2, the RL agent surpassed all basic models across all parameters, including the highest cumulative profit (as the only agent with a positive profit), maximum profit, and average profit.

5. DISCUSSION

This research focuses on the application of reinforcement learning to predict financial time series of returns. An overview of existing applications based on machine learning in financial market forecasting tasks is given. The work describes the reasons for the topic's popularity from a scientific and practical point of view. The practical part of the study is devoted to the forecast of time series. The components of makeup time series are discussed and two specific types of time series are explained: white noise and random walk. The article deals with the analysis of time series data and such research methods. The experiments emphasize the importance of pre-normalizing the data and providing an activity chart to create a financial time series forecasting program. The article also describes the technical and software development tools used to implement the software implementation of the research. It includes a detailed analysis of the construction of the specific environment that the gym interface must implement, including the action space, the observation space, and the features of the structure of the reward function. It also analyses the challenges of training an agent to work with financial data using input data and how to overcome them. The baseline models are used to compare the obtained results as benchmarks to evaluate the performance of the RL agent. The performance of all base models (results of benchmarking) and the RL agent is considered. The results are compared according to four criteria: total profit, minimum, maximum and average profit. The risk of the RL agent is also estimated by the number of negative and positive returns for each day of testing (withdrawal).

It should be noted that this issue is relevant in scientific doctrine, and therefore researchers study it in different contexts. Taking this into account, it is possible to compare the obtained results with the positions common among scientists. In particular, T.K. Balaji et al. (2021) and Y. Song et al. (2020) in their works focused on such an estimator of statistical moments of stochastic multidimensional time series as Score Machine (SM). T.K. Balaji et al. (2021) drew attention to the fact that the first-order SM(1) estimator is aimed at working on one-dimensional time series, which can be used to form their estimate. The latter provides a generalization of the characteristics of time series location parameters, including modes, medians, and means. Accordingly, a multidimensional series to which M components belong will be characterized by M first-order scores, one for each of them. In turn, Y. Song et al. (2020) analysed the operation of the second-order evaluation mechanism SM(2), the activity of which is based on two-dimensional time series. As a result, it is possible to form a correct estimate, on the basis of which to determine the features of the parameters and the features of the variance of the common series. As an example, the researchers cite covariance, as well as modal variance. At the same time, such a multidimensional series consisting of M components will have M! assessments of the second level, respectively, one for a separate pair. Analysing the results of the researchers' work, it should be noted that they have common principles with the conclusions of this article. In particular, the position that the extracted data do not in all cases correspond to the statistical, i.e. central moments of the series, coincides. Based on this, the articles conclude with a concise and informative presentation of time series statistics.

R. Tavenard et al. (2020) and also B. Lim and S. Zohren (2021) analysed the reinforcement learning model. R. Tavenard et al. (2020) gave their own definition of this educational mechanism as a separate unit of machine learning, which consists of students mastering the skills of managing a system or a specific area while increasing a numerical value that demonstrates a long-term goal. Based on this, it can be established that it supports the idea of a reduced cumulative reward signal. B. Lim and S. Zohren (2021) focused attention on approaches to the organization of reinforcement learning, which should be based on the principles of optimal management of various dynamic systems. Accordingly, in their opinion, a controller or agent should be involved in this process, which accordingly maintains the state of the system, and also ensures the realization of the reward associated with the last state transition. As a result, it can perform calculations while simultaneously controlling the signals coming back into the system. In this way, the researchers managed to reveal the mechanism that consists in the deformation of the system and its transition to a new state, which is cyclic. In turn, R. Tavenard et al. (2020) also paid attention to the ways of managing the system, in the context of the study of its management policy. Accordingly, they proved that systemic change based on reinforcement learning is appropriate to maximize the total cumulative reward. In their opinion, this approach allows for solving various kinds of problems with the help of decision-making, namely the training of agents. The latter, in turn, can optimize deferred rewards, and most importantly, assess the distant consequences of their actions. As a result, such subjects have the right to sacrifice an immediate reward and receive a long-term reward as a result. The analysis of the mentioned results of the researchers allows to conclude that the features of reinforcement learning agents allow to influence the expansion of the vectors of their use. This

conclusion resonates with the position expressed in this article, as it identifies how reinforcement learning can be implemented, based on which success can be achieved in various domains.

Unlike previous scientists, J.F. Torres et al. (2021) and A. Zeng et al. (2022) considered the feasibility of forecasting time series. J.F. Torres et al. (2021) note that such a process involves the use of a mechanism for predicting future values based on data obtained during previous observations. Taking this into account, they consider this approach to be effective and allows for achieving the set goal. A. Zeng et al. (2022) note that this type of forecasting is often identified in scientific doctrine with regression analysis. However, they disagree with this approach because she believes that the results of one or more independent time series can also affect the current value of another time series. Based on this, it should be established that time series forecasting involves comparing values within one or more-time series in different periods. This conclusion is shared with this study because it proves that time series forecasting is a separate operation and therefore has different characteristics than regression analysis. Open positions allow you to determine the specifics of stochastic time series forecasting, including using reinforcement learning. This indicates the multifacetedness of this mechanism, as well as the interdependence of its components. That is why the results of the researchers are intertwined and reveal common ideas about the implementation of reinforcement learning, as well as the provision of efficient forecasting of stochastic time series.

6. CONCLUSIONS

This work discusses the relevance of using machine learning methods, particularly reinforcement learning, for predicting financial time series. This topic has gained popularity due to its scientific and practical significance. The study provides an overview of available applications based on machine learning for financial market forecasting problems. The subject area of time series prediction has been analysed in detail, considering the structure of time series and its particular types. The importance of performing a preliminary analysis of the data that comprise the time series has been emphasized, along with the methods for conducting such an analysis. The significance of prior data normalization has also been noted. A detailed analysis of the mathematical and algorithmic support has been carried out, with particular attention given to the Proximal Policy Optimization algorithm of deep reinforcement learning. The main challenge in this work was constructing a specific environment that would correspond to the stock exchange's logic, along with analysing the problems that might arise when training an agent to work with financial data.

For the most accurate and objective assessment of the trained RL agent, four basic models were proposed and developed, built based on two strategies – buy and hold and random actions. The performance of the trained RL agent has been proven to significantly outperform the baseline models when dealing with this type of financial time series. The risk of the RL agent's work was also assessed based on the number of negative and positive returns for each day of testing (inference). It is worth noting that the training and verification were carried out with the limitation of buying and selling only one unit of the Bitcoin cryptocurrency. Therefore, further research should be carried out with an increase in this number and the possibility of using different types of cryptocurrencies and deploying this structure to work in real time. Overall, this work demonstrates the potential of reinforcement learning in financial

time series prediction and highlights the importance of careful data analysis and environment construction when working with financial data.

Acknowledgements

The study was created within the framework of the project financed by the National Research Fund of Ukraine, registered No. 30/0103 from 01.05.2023, “Methods and means of researching markers of ageing and their influence on post-ageing effects for prolonging the working period”, which is carried out at the Department of Artificial Intelligence Systems of the Institute of Computer Sciences and Information Technologies of the Lviv Polytechnic National University.

7. REFERENCES

Alieksieiev, I., Mazur, A., 2022, Methodology of financial research by stages of innovation process. *Econ., Entrepr., Manag.* **9**(1), 7-14. https://doi.org/10.56318/eem2022.01.007

Artomov, D., 2022, State and trends of urbanization in the context of the formation of a smart economy. *Econ. Bull. of Cherkasy State Tech. Univer.* **22**(4), 60-68. https://doi.org/10.24025/2306-4420.67.2022.278778

Balaji, T.K., Annavarapu, C.S.R., Bablani, A., 2021, Machine learning algorithms for social media analysis: A survey. *Comput. Sci. Rev*. **40**, 100395.

Boyko, N., Kachmaryk, V., 2022, Construction of time series forecasting models using long short-term memory networks. *Sci. Bull. Uzhhorod Univ*. **40**(1), 109-125.

Beniwal, M., Singh, A., Kumar, N., 2023, Alternative to buy-and-hold: Predicting indices direction and improving returns using a novel hybrid LSTM model. *Int. J. Artif. Intell. Tools*. **32**(7), 2350028.

Chen, J., 2020, *What is annual return? Definition and example calculation*, Retrieved from https://www.investopedia.com/terms/a/annual-return.asp

Federal Reserve Bank of Minneapolis, 2023, Consumer Price Index, 1913-, Retrieved from https://www.minneapolisfed.org/about-us/monetary-policy/inflation-calculator/consumer-price-index-1913-

Gym Documentation, 2023, Retrieved from https://www.gymlibrary.dev/

Hayes, A., 2023, *Financial markets: Role in the economy, importance, types, and examples*, Retrieved from https://www.investopedia.com/terms/f/financial-market.asp

Hayes, A., 2023, *Short selling: Definition, pros, cons, and examples*, Retrieved from https://www.investopedia.com/terms/s/shortselling.asp

Hendricks, D., Wilcox, D., 2014, *A reinforcement learning extension to the Almgren-Chriss model for optimal trade execution*. In: IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr) (pp. 457-464), London: IEEE.

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., Van Pottelbergh, T., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Koscisz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., Grosch, G., 2022, Darts: User-friendly modern machine learning for time series. *J. Mach. Learn. Res*. **23**, 1-6.

Kurhan, N., Fartushniak, O., Bezkorovaina, L., 2023, Improvement of organization and automation of commercial enterprise electronic money accounting in conditions of economy digitalization. *Econ. of Devel.* **22**(3), 8-20. https://doi.org/10.57111/econ/3.2023.08

Li, H., Dagli, C. H., Enke, D., 2007, *Short-term stock market timing prediction under reinforcement learning schemes*. In: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning (pp. 233-240), Honolulu: IEEE.

Lim, B., Zohren, S., 2021, Time-series forecasting with deep learning: A survey. *Philos. Trans. R. Soc. A.* **379**(2194), Retrieved from https://doi.org/10.1098/rsta.2020.0209

Liu, X. Y., Xiong, Z., Zhong, S., Yang, H., Walid, A., 2018, *Practical deep reinforcement learning approach for stock trading*, Retrieved from https://arxiv.org/abs/1811.07522

Loi, A., 2023, Identification of investment attraction strategies to increase the economic potential of a trading enterprise. *Econ., Entrepr., Manag.* **10**(1), 8-16. https://doi.org/10.56318/eem2023.01.008

Sadighian, J., 2020, *Extending deep reinforcement learning frameworks in cryptocurrency market making*, Retrieved from https://arxiv.org/abs/2004.06985

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017, *Proximal policy optimization algorithms*, Retrieved from https://arxiv.org/abs/1707.06347

Song, Y., Garg, S., Shi, J., Ermon, S., 2020, Sliced score matching: A scalable approach to density and score estimation. *Proc. 35th. Uncertainty. Artif. Intell. Conf.* **115**, 574-584.

Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., Payne, M., Yurchak, R., Rußwurm, M., Kolar, K., Woods, E., 2020, Tslearn, a machine learning toolkit for time series data. *J Mach. Learn. Res.* **21**(1), 4686-4691.

Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A., 2021, Deep learning for time series forecasting: A survey. *Big. Data.* **9**(1), 3-21.

Wagena, M. B., Goering, D., Collick, A. S., Bock, E., Fuka, D. R., Buda, A., Easton, Z. M., 2020, Comparison of short-term streamflow forecasting using stochastic time series, neural networks, process-based, and Bayesian models. *Environ. Model. Software*. **126**, 104669.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2022, *Are transformers effective for time series forecasting?* Retrieved from https://arxiv.org/abs/2205.13504v3