Journal of Engineering Science and Technology
Special Issue on ICIT2022 Vol. 18, No. 6 (2023) 22 - 31

ALGORITHMIC TRADING STRATEGY DEVELOPMENT USING MACHINE LEARNING

HIEW SIR LOON[1], DESHINTA ARROVA DEWI[1,*], RAJERMANI THINAKARAN[1], TRI BASUKI KURNIAWAN[2], MALATHY BATUMALAY[1]

[1]Faculty of Data Science and Information Technology, INTI International University, Malaysia
[2]Faculty of Computer Science, University of Bina Darma, Palembang, Indonesia
*Corresponding Author: tribasukikurniawan@binadarma.ac.id

Abstract

Algorithmic trading refers to using a computer application with some algorithms to identify and execute a trade at a speed and frequency that is impossible for a human trader. As it is profitable, it is one of the applications that draw the most attention from researchers. Price prediction is the main difficulty in finding and carrying out a potentially profitable trade. In this research, the author will attempt to perform cryptocurrency price predictions with three artificial intelligence techniques – Recurrent Neural Networks (RNN), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL). Machine learning (ML) library tools such as Tensorflow and Keras will be used to develop and evaluate the models. This project also studies the factors that affect the performances of models built with these three techniques and explores possible improvements to the models. The author has chosen mixed methods studies to integrate quantitative and qualitative data collection and analysis. It attempts to combine the best methodologies to integrate perspectives and create a rich picture. The author has interviewed some traders to gain insights into the critical areas in trading. As for the quantitative data collection and analysis, the selected cryptocurrency historical price data will be collected from Kaggle.

Keywords: algorithmic trading strategy, cryptocurrency, recurrent neural network, reinforcement learning, deep reinforcement learning, industrial innovation, financial inclusion

1. Introduction

Artificial Intelligence (AI) grew exponentially in the past decade, including a vast array of applications and AI breakthroughs that even surprised everyone worldwide after the epic Go matches between the AI player AlphaGo and the top human Go masters in the world [1].

Algorithmic trading refers to automated trading that consists of the use of computer programs in terms of executing and identifying trades. Algorithmic trading is one of the most in-demand technologies in the current era [2]. Algorithmic trading is a crucial tool for investigating market behavior and assessing its profitability in short and long-term scenarios. The implementation of AI in trading has gone through different stages of development.

AI can be used in trading to provide trading strategy suggestions and power automated trading systems that make predictions, choose the course of action, and execute trades. AI-based trading systems operating in the market can identify and execute trades independently, without human intervention, using AI techniques such as evolutionary computation, deep learning, and probabilistic logic [3].

Given today's interconnectedness between asset classes and geographies, AI allows for predictive capacity that quickly outpaces the power of even conventional algorithms in finance and trading. Many machine learning or AI techniques have been developed recently, addressing algorithmic trading or other problems. Many people now interact daily with systems based on machine learning [4]. Attempts have been made to adopt deep learning in predicting market trends with historical prices, while there are also attempts to relate market fluctuations with news and social events.

Since the introduction of the Deep Q-Network (DQN), most reinforcement learning research has focused on reinforcement learning with deep neural networks as function approximators. Recently, new methods are typically evaluated on environments that have become standard, such as Atari 2600 games [5]. Algorithmic trading has also gained significant market share in international financial markets in recent years as time and cost-saving automation went hand in hand with cross-market connectivity [6].

A stock market prediction has always caught the attention of many analysts and researchers [7]. Profits are the guiding force behind most investment choices. Stock market investors must know when to buy or sell stocks to maximize their investment return. However, stock market prices do not behave as a simple time series. Analysing stock market movements and price behaviors is highly challenging because of the market's dynamic, nonlinear, nonstationary, nonparametric, noisy, and chaotic nature [8].

Financial time series prediction is notoriously tricky due to the generally accepted, semi-strong market efficiency form and the high noise level [9]. The efficient market hypothesis (EMH), alternatively known as the efficient market theory, is a belief that states that share prices reflect all information and consistent alpha generation is impossible. However, with the appearance of Behavioral Finance, many financial economists believe that stock prices are at least partially predictable based on historical stock price patterns, which reinvigorate fundamental and particularly technical analysis as tools for price prediction [10].

2. Methodology

This study combined qualitative and quantitative techniques to achieve the main research goals.

2.1. Research design

This study proposes the implementation of LSTM and DQN in building BTC trading algorithms. Before that, the author will describe the architecture of each model. First, for the LSTM model, the original dataset (btc) is downloaded, pre-processed, and checked for nonsense values, missing data, and other misconceptions.

Next, a target variable (y) is generated for learning. Technical analysis variables widen the resulting dataset to form a modified dataset (btc_df) that is finally split into two (2) mutually exclusive datasets: in-sample (train_set) and out-of-sample (test_set).

The first one (in-sample) is used for training the LSTM models, as shown in Fig. 1, while the second is for evaluation only. Although passive and rule-based trading strategies do not require a dataset split, it is done anyway to bring a fair assessment of sedentary and rule-based trading strategies compared with surrogate model trading strategies [11].

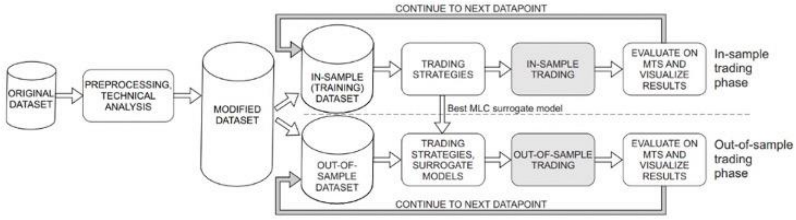

Fig. 1. Trading system LSTM design.

Next, for the DQN model, as shown in Fig. 2, when the raw data comes in, the data analysis phase is performed first to ensure that it is not extreme data. At the same time, the original data is passed into the data processing phase. Data will then be the input for the Deep Q-Network part.

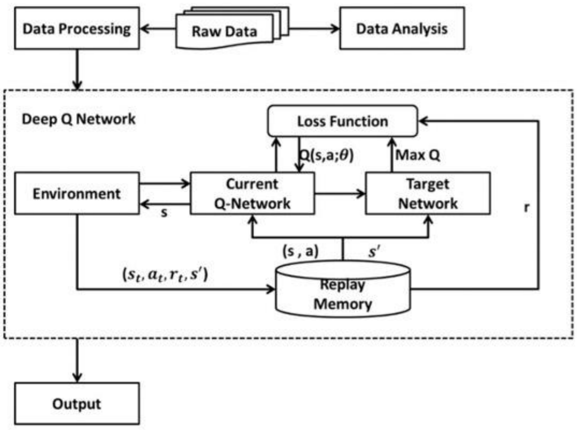

Fig. 2. Trading system with DQN model.

The training data is extracted randomly from Replay Memory, which records the actions (a), rewards (r), and results of the following state (s, a, r, s 0). As the Environment changes, Networks will update their parameters regularly, and Replay Memory will change accordingly.

The Loss function is the result of subtracting the value of Q in the Target-Network from the value of Q in the Current qnetwork. The values between modules are changed iteratively until the optimal value of Q is achieved, and the output operation is carried out [12].

2.2. Quantitative method

Quantitative data collection and analysis are required to create AI models for price prediction and portfolio management modules for cryptocurrencies. The selected cryptocurrency, Bitcoin, and historical price data will be collected from Kaggle. The data from established exchanges, from Jan 2012 to March 2021, is available from Kaggle.

Kaggle allows the dataset to be exported in CSV format. The sample selected is bitcoin exchanges from Jan 2012 to March 2021. The data included are minute updates of Open, High, Low, and Close (OHLC), Volume traded against BTC & Volume traded against other currencies, weighted BTC price, and Timestamps with NaNs if null value.

NaNs are expected due to technical issues on the exchange platform or other unforeseen data reporting or gathering problems if possible further data will be created with a function to record technical analysis (TA) and graph for pattern studies.

Below are the data columns that will be extracted from the dataset, which the author considers these fields are essential in building the price prediction and portfolio management models.

- Timestamp
- Open
- High
- Low
- Close
- Volume_(BTC)

3. Result and Discussion

The dataset contains 4,857,377 rows with 1,243,608 missing values, for a total of 4,857,377 rows. It is removed from the dataset any missing values that may exist. This study will predict the next day's Closing price of BTC following the methods described above.

There was no significant fluctuation of BTC before 2017. Thus, this study will only use the timestamp from January 2017 to March. As described before, the dataset will be split into training and test sets. The author takes the "‘2021-01-0" timestamp as a date split. Hence, the test set contains 90 days, and Fig. 3 below shows the train and test data.

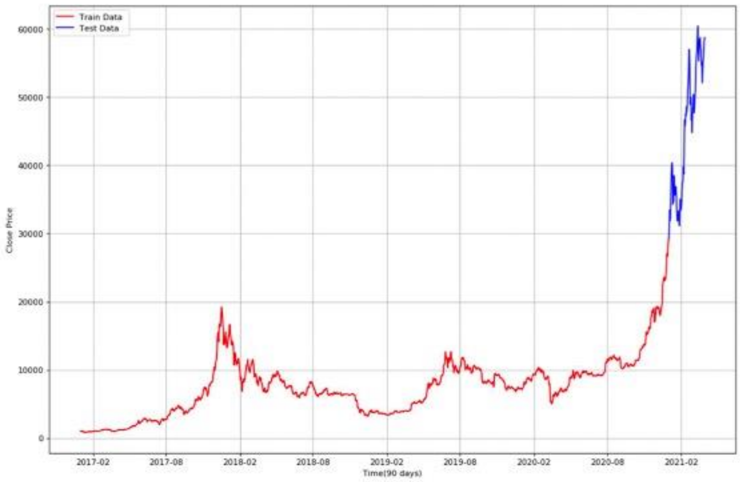

Fig. 3. Data graph from 2017-2021

3.1. LSTM result

The parameter setting of the LSTM model is shown in Table 1.

Table 1. Result accuracy split ratio.

Parameter	Value
LSTM layer activation function	sigmoid
Optimizer	Ada
Loss function	Binary cross entropy
Metrics	accuracy
Epoch	50
Batch size	20
Min learning rate	0.000001

According to the parameter setting and model summary, the LSTM model is constructed as follows in Table 2: the main input layer of training set data is a three-dimensional data vector (None, 7, 6), indicating that the LSTM model used the previous week's data as its input. Six (6) features of the BTC dataset are used for the input, i.e., open, high, low, close, Volume in currency, and Volume in BTC.

Table 2. LTSM model.

Layer (type)	Output Shape	Param #
main_input (InputLayer)	(None, 7, 6)	0
gaussian_noise (GaussianNoise)	(None, 7, 6)	0
flatten (Flatten)	(None, 42)	0
batch_normalization (BatchNo)	(None, 42)	168
dropout (Dropout)	(None, 42)	0
out (Dense)	(None, 1)	43

The accuracy and loss of forecasting for the LSTM model are shown in Figs. 4 and 5, respectively. The LSTM model generates an accuracy of 0.62.

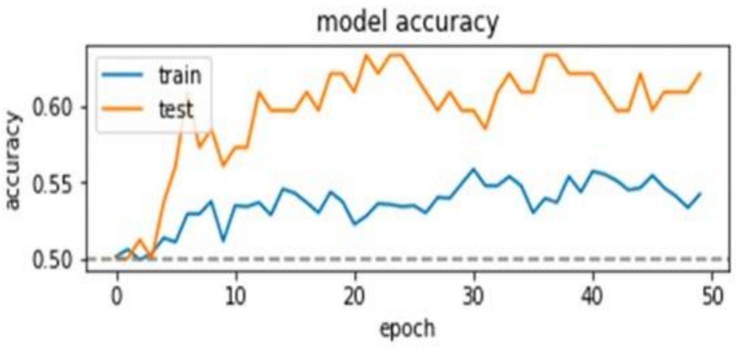

Fig. 4. Data Model Accuracy

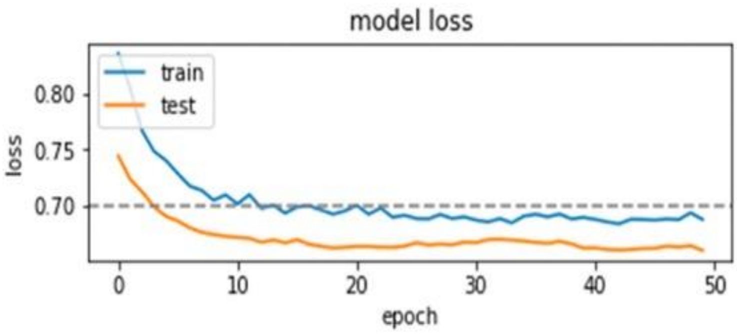

Fig. 5. Data Model Accuracy

When building a trading system, the next step is to specify a trading strategy that uses deep learning model predictions as input and outputs actual buy/sell orders. A good trading strategy should take full advantage of the model's predictions by understanding the prediction's essence and considering the existing positions when generating trade orders [13]. The trading model proposed in this study is designed to predict tomorrow's returns.

Moreover, this study develops "‘Market Intraday Portfolio", an environment where trading strategies can be tested. A key advantage of having a market simulator is that we can back-test a trading strategy to see how it would have performed in the past. Back-testing is a critical difference between a traditional investment management process and a quantitative investment process [14]. Not only can we derive complete detail of the performance of a strategy, such as daily returns, position changes, and cash changes, but also back-testing

The trading strategy will generate 1 as Buy signal, -1 as Sell signal, and 0 as Hold signal. With the initial capital of $100,000, the BTC trading strategy is represented below in Figs. 6 and 8.

Figure 7 shows the performance of the trading system on BTC that the LSTM model produces. The system trades from January March 1st to 30th, 2021, representing three months out of the sample test. The returns of this trading strategy are shown in Table 3.

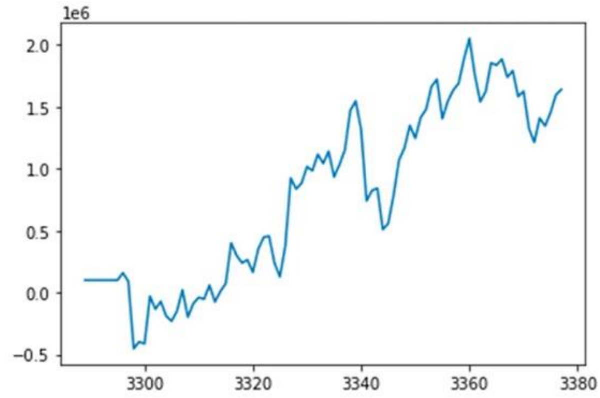

Fig. 6. BTC Trading

Fig. 7. BTC Trading continued

Table 3. Data resample.

Time stamp	price_diff	profit	total	returns
3289	1471.235229	0.000000	1.000000e+05	NaN
3290	2710.854854	0.000000	1.000000e+05	0.000000
3291	-1610.974868	-0.000000	1.000000e+05	0.000000
3292	417.699806	0.000000	1.000000e+05	0.000000
…	…	…	…	…
3376	1434.470492	143447.049240	1.597249e+06	0.098670
3377	430.374376	43037.437631	1.640286e+06	0.026945
3378	NaN	NaN	Nan	0.000000

3.2. DQN result

In the DQN method, reinforcement learning algorithms were used to simulate trading. Figure 8 shows an overview of the time-market value of the BTC in the period January 2017 to March 2021.

Fig. 8. DQN result graph

Next, Fig. 9 shows the Reward and Loss functions of the selected stock after the training by DQN models. The loss function estimates the degree of inconsistency between the model's predicted value and the actual value, which is mapped by an event (an element in a sample dimension). The experiment compares the loss functions of the three DRL models to measure the economic benefits of this model, as shown in Fig. 10.

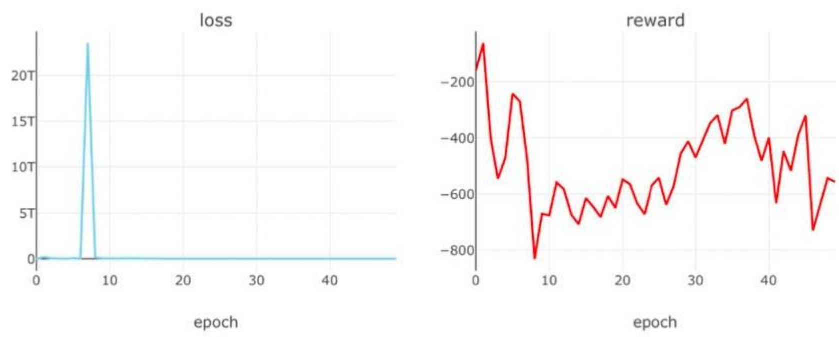

Fig. 9. DQN loss and reward result

Fig. 10. DQN Result Continued

As seen in Fig. 10, even though the DQN model generates positive profit in the train set, the profit of the test set is negative. Furthermore, the DQN model does not generate positive rewards. If the agent's objective were only to reach the goal, it would be enough to put the positive reward in the state space. However, the agent aims to avoid obstacles and reach the goal. We might put some negative rewards in the state space.

A negative reward represents a bad result and should be avoided by the agent. Because the agent acts to increase the sum of rewards, the states with negative rewards are not of interest to the agent [15]. In conventional Q-learning, a positive worth can be propagated around the goal area, but a negative value cannot be reproduced. The highest Q-value of the following state st+1 is selected in the equation [15].

Successful trading strategies should not only take full advantage of the model's forecasts by fully comprehending the content of the prediction, but they should also consider the current positions when generating trade orders. It is advantageous to use a market simulator because it allows us to back-test a trading strategy to see how it would have performed in the past. An important distinction between a traditional investment management strategy and a quantitative investing method is using back-testing in the investment process. While back-testing allows us to obtain a detailed picture of a strategy's performance, including daily returns, position changes, and cash flow changes, we can also experiment with several variants of the original strategy to refine and improve it.

As a result, large and small organizations and the machine learning system may benefit from improving their systems and goods. As a result, companies may spend more money on systems than traditional tools because the new design offers advantages over the old one.

Due to this finding, companies are advised not to rely solely on old systems but instead incorporate them into their new plans. By utilizing a variety of channels, the company can gain the trust it needs and project a positive image to the public at large. Consumers will trust a brand more if it has earned their credibility.

4. Conclusion

Algorithmic trading is the practice of using a computer program to choose and carry out a deal at a rate and frequency that is higher than that of a human trader. This application's potential for financial gain has drawn the attention of researchers in particular. When it comes to finding and carrying out a potentially profitable transaction, price prediction is the most challenging task. The ultimate aim of this algorithmic trading challenge is to create an algorithm that can learn to trade profitably, so it doesn't matter which strategy is employed. Automated trading systems must include statistical analysis of market behavior and methods for calculating profitability in both short- and long-term scenarios (ATS).

The stock level prediction and portfolio management strategies were thoroughly examined in this study as two ways to handle the algorithmic trading issue. Recurrent neural networks, deep reinforcement learning, and reinforcement learning were used to implement the trading strategies. Artificial intelligence models were created, and the results were contrasted. Every objective, including comparing and evaluating various machine learning techniques for trading algorithms, was met or even beyond.

References

1. Toosi, A.; Bottino, A.G.; Saboury, B.; Siegel, E.; and Rahmim, A. (2021). A brief history of AI: how to prevent another winter (a critical review). *PET Clinics*, 16(4), 449-469.
2. Miller, R.K.; and Walker, T.C. (1988). *Artificial intelligence applications in engineering*. SEAI Technical Publications.
3. Millea, A. (2021). Deep reinforcement learning for trading - A critical survey. *Data*, 6(11), 119.
4. Alzubi, J.; Nayyar, A.; and Kumar, A. (2018). Machine learning from theory to algorithms: an overview. *Journal of physics: Conference Series*, 1142, 12012.
5. Obando-Ceron, J.S.; and Castro, P.S. (2021). Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. *Proceedings of the 38th International Conference on Machine Learning*, PMLR 139:1373-1383.
6. Chen, S.H.; Kaboudan, M.; and Du, Y.R. (2018). *The Oxford handbook of computational economics and finance*. Oxford University Press.
7. Shah, D.; Isah, H.; and Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2), 26.
8. Guerard Jr, J.B. (2013). *Introduction to financial forecasting in investment analysis*. Springer Science and Business Media.
9. Shen, J.; and Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, 7(1), 1-33.
10. Malkiel, B.G. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1), 59-82.
11. Fister, D.; Mun, J.C.; Jagrič, V., and Jagrič, T. (2019). Deep learning for stock market trading: a superior trading strategy? *Neural Network World*, 29(3), 151-171.
12. Lei, Y.; Peng. Q.; and Shen, Y. (2020). Deep learning for algorithmic trading: enhancing MACD strategy. *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*. Tianjin China, 51-57.
13. Lu, N. (2016). A machine learning approach to automated trading. Boston, MA, USA: Boston College Computer Science Senior, 7, 171-180.
14. Chan, E.P. (2021). *Quantitative trading: How to build your own algorithmic trading business*. John Wiley and Sons.
15. Fuchida, T.; Aung, K.T.; and Sakuragi, A. (2010). A study of Q-learning considering negative rewards. *Artificial Life and Robotics*, 15(3), 351-354.