SIT 746 Task 2.2D: Completed Research Methodology

In SIT 723 the research questions were:

What is the optimal set of features (technical indicators) for constructing the state space in a stock trading DQN agent?

How do various DQN variations (e.g., DDQN, PER, dueling networks, noisy nets, multi-step learning, C51) compare in performance for stock trading applications?

What is the impact of different reward functions (e.g., profit return vs. Sharpe ratio) on the learning and decision-making process of the agent?

Can an ensemble approach, built upon the insights from individual DQN performance, generalize across multiple market regimes and outperform standard strategies?

SIT 723 Limitations:

The issue that arose with these research questions were that they were too open ended and involved conducting a factorial design study: 15 different technical indicator combinations, followed by 27 different DQN combinations and 5 different reward structures. This resulted in an exhaustive experimental phase which required most of the time spent on doing experiments instead of research. Essentially it was a comparative study to find the optimal combination. However, there is also the possibility that switching the order, for example testing reward function first, could result in a different optimal combination.

Feedback from my supervisor was that the thesis in SIT 723 was good for a first time doing a research project, but for SIT746 I need to follow a more structured approach so my research would have a higher chance of being published in top publications like Neural IPS.

SIT 746 Research Question:

Utilising my supervisor's feedback and expanding on my research we came up with a better research question:

How can we improve the out-of-sample generalisation and training stability of DQN-based stock trading agents?

Building on value-based, off-policy DRL, I focus exclusively on DQN and its advanced variants (DDQN, C51, PER) because stock trading is inherently a discrete decision problem (buy / hold / sell), and DQN's replay buffer naturally accommodates the nonstationary of price series. In contrast, actor-critic or model-based methods, while powerful for continuous controls or model learning, tend to introduce additional variance or modeling bias, which can destabilize financial agents. By concentrating on pure DQN, I can directly tackle stability and sample-efficiency challenges in the setting where DQN is already a proven baseline.

SIT 746 Research Method Approach:

Discussing with my supervisor we formulated a much more impactful research approach, in which I find a top tier publication from a Q1 journal or A* conference directly related to my research focus. Then using their approach as a baseline, I can identify any gaps and use existing literature to build a more robust method to expand upon their work and improve it, hence making a notable contribution to the field.

SIT 746 Literature Review Summary

From the completed literature review I have identified these notable gaps:

1. No standard baseline for the state space when it comes to stock trading using DRL.
2. Lack of feature engineering for the state space in DQN - stock trading.
3. No use of C51 distributional RL and lack of utilising the more advanced DQN variants.

The literature review identified that using the standard candlestick-chart time series, which comprises of Open, High, Close, Low and Volume (OHCLV) is a suitable baseline for DRL-stock trading. Many authors have utilised technical indicator time series based off the OHCLV data, but there is no general consensus on which configuration is optimal and a plethora to choose from. There is also the issue of multicollinearity with many indicators being highly correlated. Some authors have shown that just using OHLCV prevents the curse of dimensionality and is enough information to feed into the neural network. Shi et al (2021) mentions the Efficient Market Hypothesis, which states all the information about a stock lies in their price, hence deriving extra features based on OHCLV is redundant. So in my work I will formalise the use of OHCLV as a suitable baseline for future researchers to work upon.

The literature then showed that the use of the Kalman filter to de-noise the OHLCV data offers the best performance in finance deep learning tasks. However, the Kalman filter has not been applied in DQN -stock trading to my knowledge. Stock time series data is often very noisy.

The best sequential architecture that has been proven to work well in financial DRL task is the CNN. So the first part of the neural network will consist of CNN layers. From SIT 723 the best performing DQN structure was the Multi-step + PER + C51 (MPC) combination. This agrees with the ablation studies from Hessel et al (2017).

Benchmark Paper Method

Cui et al (2023) is the benchmark paper I found that is from a Q1 journal and directly aligns with my research focus. In their paper they proposed a multi-scaled CNN (MS-CNN) + DDQN architecture for DRL-stock trading. Their state space was OHLCV and they used a Sharpe Ratio reward function. Their action space is also buy/hold/sell as in my research.

Their method was to standardise the state space then feed the data in windows of 20 days into their MS-CNN + DDQN. They used data from 2007 Jan to 2017 Dec for training and Jan 2018 to Dec 20 for testing. They used Apple, General Electric (GE) and DJI as there chosen stocks. They chose Apple as it was in a bull market, GE was in a bear market and DJI was in a side-ways market. This meant they can show how their method performs in each type of market.

They used profit, Sharpe ratio and annualised return as their evaluation metrics. They started with a portfolio of 500,000, used a transaction cost of 0.1% and the Adam optimizer with learning rate 0.001 and trained for 100 epochs. All other hyperparameters used for the algorithm were not mentioned and there is no publicly available code. Their method beat 5 other baseline methods.

Benchmark paper flaws

While the MS-CNN method they proposed is highly sophisticated and analyses time patterns over multiple frames, it contains roughly 10 different CNN layers before it reaches the 2 DQN layers. Each step only 20x5 (100) data points are feed. This means there is roughly 1000s of parameters needed to estimate at each step based on only 100 data points. There is a high risk of overfitting. The paper also shows no validation graph to show their reward learning

does not show signs of overfitting. Since there is no code available it is not feasible to replicate their design to show how their method performs on validation data. However, this highlights the need for its inclusion, to better justify the choice of their network. Also, they only use 3 stocks. To test the effectiveness of their approach they should perform analysis on more stocks or other assets such as cryptocurrency.

My Approach

Building on their limitations, first I will utilise the Kalman filter to preprocess my data before standardizing it. From the literature review I saw that utilising two CNN layers was able to produce superior performance, so I will feed the state space into the dual layered CNN before it goes into the MPC DQN.

For the reward function, I cannot use Sharpe Ratio as they did. When using multi-step learning the multi-step reward accumulates the rewards over n steps (10 days here) as $r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^n r_{t+n}$. In the context of financial tasks, if r is the Share Ratio, then the accumulated reward makes no sense in the context of finance. The Sharpe Ratio is meant to be measured in one instance, so can be used if multi-step is not being used. However, using a log return, accumulating over n steps (10 days) gets the accumulated profit over 10 days' worth of trading so the agent can see how well the choice of actions performs over 10 days instead of one.

The action space remains the same, the agent can either buy, hold or sell the stock.

The training and test data will be the same as in the benchmark paper. This way I can report their results as a benchmark against mine, since their code is unavailable. However, from sit 723 I saw training on the whole train period results in overfitting as the agent can memorise the data. Hence, I will again utilise random windows of 315 steps (1.5 years of trading data). From hyperparameter tuning this result offered the best performance in terms of speed and profit. I will use the test period as validation data to show there is no signs of overfitting with my approach.

For evaluation I will use the same metrics, profit, Sharpe Ratio and annual return. For presentation of results, I will showcase the buy and hold method (buy stock day 1 and hold until the end) with the benchmark results along side mine.

For generalisation studies, to show how well the algorithm generalises, I will test the algorithm on the top 5 cryptocurrencies in today's market: Bitcoin, Ethereum, XRP, Binance Coin and Solana. As the cryptocurrency market is not as mature as the stock market, due to the liquidity assumption (where my trades will have no impact on the price) I must use the top cryptocurrency coins as they are the only coins with enough market capitalization to satisfy the liquidity assumption. The training period will be 2018 Jan to 2023 Dec. The test period will be Jan 2024 to Jun 2025. This training period was chosen as the crypto market became mainstream from the end of 2017.

I will make sure to show the validation reward learning curves as well.

Lastly, I will conduct ablation studies on removing the Kalman filter and MPC (separately) to show the effectiveness of the Kalman filter and MPC individually. Showcasing while both MPC and Kalman filter is needed.

The benefit of this approach is that the experimental phase will be much quicker than in SIT 723. Since I have a properly defined method, it will take 1-2 weeks top. When reporting results, I will do 10 runs to get a confidence interval of the results and graphically shows this.

I will get the portfolio time series for each run and average it out and using the standard error create a shaded channel to show the range of portfolio against the buy and hold method.

Research Timeline

Data is collected freely from Yahoo Finance, which poses no **ethical** concerns. At the time of writing this research methodology the literature review is done. The only tasks left are conducting the experiments, collecting the results and analysing them. Then finally writing up the final thesis paper. The timeline to complete SIT 746 will be:

Week 7: Conduct all experiments in Python using the forementioned approach. I will utilise PyTorch for the neural network as done in the benchmark paper.

Week 8: Collect all results, graphs and summarise the results and their implications to see if I have achieved what I set out to.

Week 9: Write up the draft thesis with the following structure:

1. Abstract
2. Introduction (includes motivation and summary of my contributions)
3. Background
4. Literature Review
5. Research Methodology
6. Results
7. Discussion
8. Conclusion
9. References
10. Appendix

Week 10: Using supervisor's feedback finalise final draft of thesis for submission.

Milestones and Sustainability

Milestone 1: Complete all experiments including ablation experiments.

Milestone 2: Create a full results summary and discussion of the results.

Milestone 3: Create first draft of the full thesis

Milestone 4: Submit the full thesis

To maximize the long-term impact and relevance of this work on DQN-based stock trading agents, I will adopt the following strategies:

1. **Open-Source Release & Documentation**
   • Publish all code and data on a public GitHub repository.
2. **Dissemination & Community Engagement**
   • Present findings at academic venues if my paper is published.
3. **Sustainability & Extension**
   • Establish a roadmap for future enhancements in the discussion.