Exploiting SEC Insider Trading Form 4 Filings: A Dueling Double Deep Q Neural Network-based Reinforcement Learning Approach

Eric Burton Samuel Martin
eric.burton.martin@colostate.edu
Department of Computer Science
Fort Collins, CO, USA

Tarun Sai Pamulapati
tarunsai.pamulapati@colostate.edu
Department of Computer Science
Fort Collins, CO, USA

ABSTRACT

This paper aims to identify the potential of exploiting legal insider trading information from SEC Form 4 filings using advanced reinforcement learning techniques. By harnessing a Dueling Double Deep Q-Network (DDDQN), we investigate whether residual market effect following insider trades can be identified and utilized for financial gain within a 14-day window post-disclosure. The methodology consists of extensive data preprocessing, real-time data integration via Apache Storm, financial feature engineering, and the development of a curriculum-based AI trading agent. The unique combination of Double Deep Q-Networks and Dueling Networks, enhanced by Prioritized Experience Replay was chosen as it has been shown to handle complex, dynamic environments similar to financial markets. Preliminary results following Curriculum 1 indicate that although the agent is demonstrating potential, significant refinements in risk management and decision-making need to be addressed in the subsequent curricula.

CCS CONCEPTS

• **Computing methodologies** → **Intelligent agents**; **Machine learning**; Distributed computing methodologies; • **Applied computing** → Forecasting.

KEYWORDS

Reinforcement learning, Q-learning, Neural networks, Artificial intelligence, Distributed Learning, Big Data, Apache Storm, Data Mining, Trading Strategy, Stock Market, Finance, Time Series, Dueling Networks

ACM Reference Format:
Eric Burton Samuel Martin and Tarun Sai Pamulapati. 2023. Exploiting SEC Insider Trading Form 4 Filings: A Dueling Double Deep Q Neural Network-based Reinforcement Learning Approach. In *Big Data, January 18 – May 9, 2024, Fort Collins, CO, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The financial markets are a realm where transparency and secrecy seem to be at odds. In the aftermath of the stock market crash of 1929, the U.S. Securities and Exchange Commission (SEC) was established with a mission to restore investor confidence and ensure a level playing field by mandating fair disclosure of financial information. [16] This initiative was intended to democratize access to financial data, allowing every investor to receive data at the same time and thus have equal opportunity to make decisions. However, despite the intention of producing a fair market, a significant disparity remains in the utility of this information due to after-market trading. The timing of financial filings submitted to the SEC are almost always synchronized with market close which creates a window for Wall Street's seasoned players to act on these insights before the broader public has a chance. This latency in the ability to act immediately on filing perpetuates the advantage for the financial elite, leaving everyday investors a step behind. So much for a 'fair market'.

This led us to wonder, is there a way to exploit the legal insider trading knowledge of the elite to our advantage, even while being a step behind? Well, the SEC requires all company insiders and shareholders owning 10% or more of a company to complete a Form 4 – Insider Trading document within two days of an insider trade. Although required to be filed two days after the trade, they can act as a beacon in the dark since they unveil the buy and sell activities of those with the most intimate knowledge of a company's prospects. For example, if Elon Musk decided to purchase a large share of a specific mining company, he would be required to file an insider trading form within two days of the purchase, disclosing the number of shares purchased, when he bought, and at what price. While the transactions reported will have already influenced the market by the time they are publicly disclosed, we hypothesize that a window of opportunity may remain open for a short time. Utilizing deep reinforcement learning, we attempt to identify these opportunities by unveiling patterns between Form 4 filings and their corresponding markets that remain exploitable in the weeks after their disclosure.

In order to achieve this, we begin with the historical SEC Insider Transaction Data Sets hosted by the SEC which contain all pertinent filings since January 2006. [15] We then reached out to Alpha Vantage, a provider of real-time and historical financial market data through a set of powerful and developer-friendly data APIs, and were granted a free academic key providing 100 requests/minute. [3] Utilizing these resources, we set up an Apache Storm topology to parse each cleaned Form 4 filing and simultaneously retrieve corresponding historical financial data for each transaction. This approach, involving millions of transactions, proved highly suitable for Apache Storm's capabilities. Following the initial training phase,

CS535 '24, January 18 – May 9, 2024, Fort Collins, CO, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

our agent was readily equipped to integrate with the SEC's Real Simple Streams (RSS) feeds for real-time functionality.

Once the data wrangling was complete we cleaned the data and performed feature engineering to append various financial features used by day traders to provide the agent more options for informed decision-making. The AI agent's base architecture utilizes a unique combination of reinforcement learning techniques, Double Deep Q-Networks (DDQN), and Dueling Networks in order to attempt to best handle the volatile and unpredictable nature of the stock markets. The Agent was trained in a distributed fashion on two Nvidia A100 39gb GPUs which is designed specifically for A.I and M.L applications.

If our hypothesis is positive, this project could be used by the common shareholder to help exploit the market adjustments that follow insider trading activity. As this paper is not going to reach a large volume of readers, the exploits discovered should be viable. If a large number of people attempt to follow the patterns elucidated, the act of following the same strategy will nullify the strategy. But with our methodology, one can elucidate patterns following other SEC reports and build on this approach.

2 CONJECTURE

Our paper hypothesizes that despite the immediate market adjustments following legal insider transactions, residual effects may linger due to the effect on other investors' reactions in this volatile moment, offering a fertile ground for short-term gains within a 14-day window following the transaction.

3 DATASETS

The analysis leverages two primary datasets:

(1) **U.S. SEC Insider Transactions Data Sets**
- The dataset, as of the last access, contains approximately 790.97 MB and 4,887,497 entries from January 2006 to the present, sourced from the U.S. Securities and Exchange Commission [16]. The dataset contains Forms 3, 4, and 5 filings in an "as-filed" format meaning data cleansing is required. [**?**]

(2) **Alpha Vantage API:**
- Alpha Vantage provides free real-time and historical financial data with intra-day granularity and was kind enough to provide us with a free academic API key with a request rate of 100 requests per minute, suitable for our project needs. [3].

4 DATA WRANGLING

This section discusses the methodologies employed in acquiring, cleaning, transforming, engineering, and integrating raw data into a structured format suitable for our agent.

4.1 Data Preprocessing of SEC Data

The Raw SEC Insider Transactions Dataset required the following data cleansing steps:

4.1.1 ***Initial Data Handling.*** The SEC dataset consisted of tab-separated value files extracted from 70 zip archives. These archives contained 8 tab separated value files each containing details like transactions and holdings. We filtered specifically for Form 4 filings and merged them to form our base dataset based on the unique identifier `ACCESSION_NUMBER`.

4.1.2 ***Data Cleaning.***
- Prefixing column names across tables based on their respective files to avoid column name conflicts after merging.
- Dropping rows with any missing values in critical columns like `ISSUERTRADINGSYMBOL` (stock ticker) and transaction details, ensuring the dataset's integrity.
- Handled missing data by excluding columns with more than 10% missing values and removing rows with missing entries leading to a dataset with no missing values.
- Addressed malformed dates and issuer trading symbols that are expected in an "as-filed" dataset.

4.1.3 ***Feature Selection.*** Only a subset of features was utilized for analysis, focusing on transaction dates and codes, shares involved, prices per share, and holding details after transactions.

4.1.4 ***Data Transformation.*** The 18 transaction codes representing the type of trade along with other categorical values were encoded using one-hot encoding to prepare the data for machine learning algorithms.

4.2 Data Integration Using Apache Storm

We utilized Apache Storm to merge the cleaned SEC data with financial data from the Alpha Vantage API, producing CSV files containing each SEC filing with its corresponding two-month window of 30-minute granularity financial data.

4.2.1 ***Topology Configuration.*** Our topology consists of two main parallel components, see Figure 1:

Spouts. Tasked with gathering and emitting SEC data with its corresponding financial data:
- Cleaned SEC data is divided into two halves to facilitate parallel processing, with each part handled by a separate instance.
- Each spout fetches two months of 30-minute granularity financial data surrounding the SEC Form 4 filing data at a rate limit of 50 requests per minute.

Bolts. Handle the bulk of data processing:
- `CollectingBolt` retrieves financial data for a two-month window surrounding each SEC filing's date. It ensures that financial data spans from one month prior to the filing date to one month after.
- Financial data fetched includes intra-day details such as open, high, low, close prices, and trading volumes, captured at 30-minute intervals.
- Each bolt aggregates the SEC filing data with its corresponding financial data, then writes the combined data to a CSV file.

4.3 Feature Engineering

To enrich the input dataset we have engineered a set of technical indicators beyond simple price data. The following indicators are typically used in day trading and will provide the agent with a

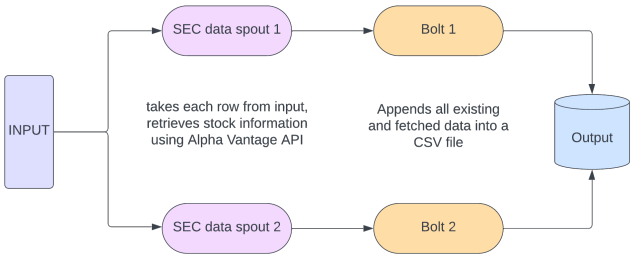


Figure 1: Apache Storm Topology Overview. The input consists of the cleaned SEC dataset

greater ability to discern more complex trading patterns and make informed trading decisions.

(1) **Exponential Moving Average (EMA):** The Exponential Moving Average (EMA) is a technical indicator that plots the average of an asset over a specific period assigning greater weight and significance to recent data points. [2]. Ths data is used to generate a buy signal (1) when the short-term EMA crosses above the long-term EMA and a sell signal (-1) when the opposite occurs.

(2) **Moving Average Convergence Divergence (MACD):** Measures stock price momentum and trends through 26-day and 12-day EMAs. The MACD line is the difference between these EMAs, supplemented by a nine-day EMA signal line. A buy signal occurs when the MACD crosses above signal line and a sell when it dips below. [6].

$$\text{MACD crossover} = \text{12-day EMA} - \text{26-day EMA}$$

$$\text{Signal line} = \text{9-day EMA of MACD crossover}$$

(3) **Volume-Weighted Average Price (VWAP) Position:** Averages stock price based on volume, recalculated daily to pinpoint trend direction. VWAP is calculated as: [5].

$$\text{VWAP} = \frac{\sum(\text{Volume} \times \text{Typical Price})}{\sum(\text{Volume})}$$

where Volume represents the number of stocks bought or sold as mentioned in SEC filings, and Typical Price is calculated as:

$$\text{Typical Price} = \frac{(\text{high} + \text{low} + \text{close})}{3}$$

(4) **Bollinger Bands (upperBB, lowerBB):** Shows price volatility and relative highs/lows using three bands, a central 20-day moving average and upper/lower bands based on standard deviation. A buy signal is when the price drops below the lower band and a sell is when it crosses above the top band. [9].

$$\text{Middle Band} = \text{20-day Simple Moving Average} \quad (1)$$

$$\text{Upper Band} = \text{Middle Band} + 2 \times (\text{20-day stdDev}) \quad (2)$$

$$\text{Lower Band} = \text{Middle Band} - 2 \times (\text{20-day stdDev}) \quad (3)$$

(5) **Rate of Change Indicator (ROC):** A momentum indicator used to calculate the percent change in price from one period to the next, comparing the current closing price with that from 12 days prior [1].

$$\text{ROC} = \left(\frac{\text{Close Price} - \text{Previous Price}}{\text{Previous Price}} \right) \times 100$$

Previous Price = Closing price 12 days ago.

5 MODEL ARCHITECTURE

This project utilized PyTorch [11] due to its ease of use and its distributed computing and GPU acceleration capabilities and leveraged 2 NVIDIA A100 GPUs within a high-performance computing (HPC) cluster at Colorado State University.

5.1 Dueling Double Deep Q-Network (DDDQN)

The Dueling Double Deep Q-Network (DDDQN) as seen in Figure 2 is an advanced variant of the standard Deep Q-Network (DQN) [10] that improves the performance and stability of reinforcement learning algorithms in complex decision-making contexts, like financial markets. [18] It combines two key techniques: Dueling Architecture and Double Q-Learning.

5.1.1 ***Dueling Architecture.*** The dueling architecture introduces two separate streams within the Q-network: a value stream $V(s)$ and an advantage stream $A(s, a)$. The value stream estimates the value of being in a particular state, while the advantage stream estimates the advantage of taking a specific action in that state. [17] These streams are combined to produce the Q-values $Q(s, a)$ as follows:

$$Q(s, a) = V(s) + A(s, a) - \text{mean}(A(s, a))$$

This separation allows the DDDQN to discern the value of states without needing to evaluate the effect of each action in every state, which is particularly beneficial for analyzing financial data.

5.1.2 ***Double Deep Q-Learning.*** Double Q-Learning addresses the overestimation of Q-values seen in standard Q-Learning by using two separate networks: a target network and an online network. The online network selects actions, and the target network evaluates these actions, which helps mitigate the overestimation issue and leads to better convergence and stability. [17] The target network's weights are periodically updated to the weights of the online network to maintain a stable learning process, see Figure 3.

5.1.3 ***Prioritized Experience Replay.*** As shown in the paper by Google DeepMind scholars [13], prioritized experience replay (PER) improves upon their previous work involving standard experience replay [12] by assigning a priority to each experience based on the magnitude of the temporal difference (TD) error, which is the difference between the predicted Q-value and the target Q-value. This prioritization helps the agent focus on more significant experiences, which often accelerates learning and enhances performance in complex learning environments, and helps maintain a balance between exploring new strategies and exploiting known strategies which is great for trading scenarios.

5.2 Model Size

The online and target network DDDQN models are each 6.9 megabytes with a total model size of 12.18 megabytes

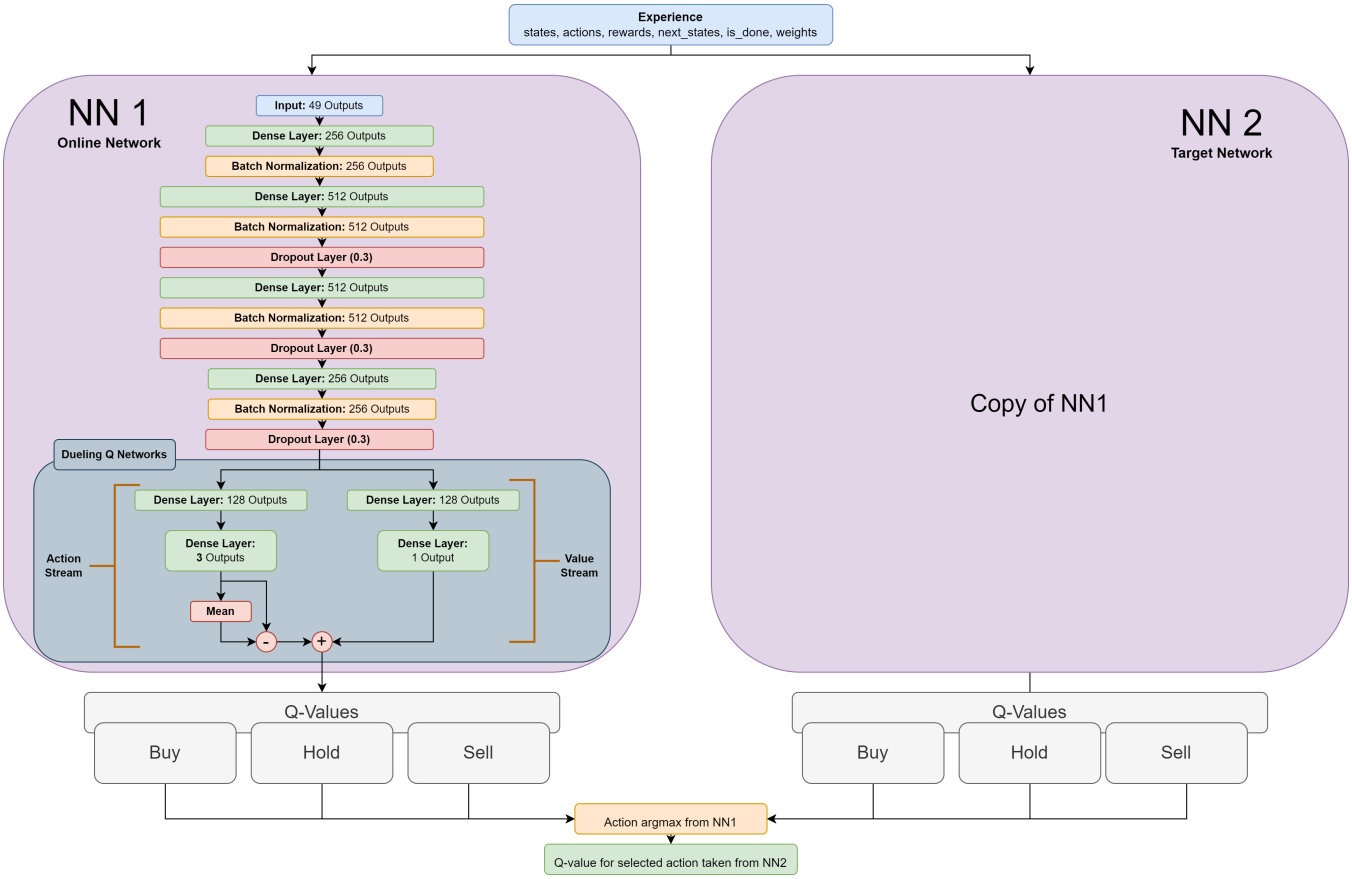
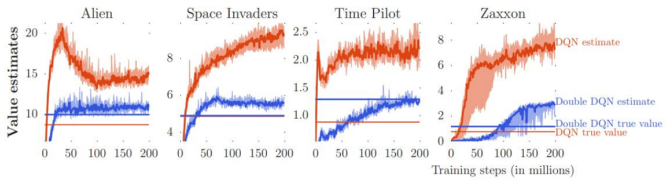


Figure 2: Model diagram of the DDDQN utilized by the trading agent inspired by the thesis by David Gallo. [7]



(Wang et al., 2015)

Figure 3: DQN and DDQN estimation values versus true values [17]

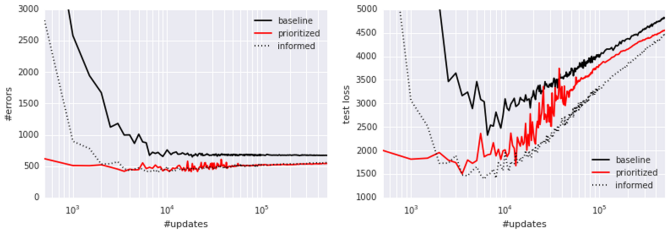


Figure 4: PER used on MNIST [8] data outperforms the baseline and approaches the performance of the informed baseline in terms of generalization [13]

5.3 Agent Architecture and Reward System

The `Agent` class is the most crucial component of our reinforcement learning framework. Where the DDDQN is the brain, the Agent is the body. We are proposing a curriculum learning-based approach based on work done by Bengio,Louradour, Collobet, and Weston. [4] This approach helps the agent learn how to manage new scenarios and learn incrementally as seen in Figure 5.

5.3.1 ***Action Space.*** The agent's decision-making process involves selecting actions based on the current state, previous experiences, exploration rate, and curriculum logic. The following are the possible actions in Curriculum 1.

- **Buy:** If the current timestamp is after the Form 4 filing data, purchase a share.
- **Sell:** If securities are currently owned, liquidate all positions.
- **Hold:** Maintain current positions.
- **Do Nothing:** If no securities are held, wait.

Actions are determined by the policy network within the DDDQN architecture, which evaluates the expected long-term rewards of each action from the current state.

5.3.2 ***Reward Mechanism.*** The reward system is the most difficult aspect of the agent to design. Here we present the reward system for Curriculum 1.

- **Unrealized Gains:** When the agent is holding assets and the current portfolio value (calculated as (current close price × number of shares held) - (price paid for all shares held)) increases, a positive reward is granted. This incentivizes the agent to grow the portfolio value before executing a sell action. Negative changes in portfolio value apply a small penalty to prevent holding for too long.
- **Realized Gains:** Positive and negative rewards are assigned after a sell action based on the profit percentage gained or lost ((total value of sold assets - price paid for all shares held) / price paid for all shares held).
- **Risk-Adjusted Returns:** To help the agent better learn to identify better trades, the Sharpe Ratio [14], representing risk-adjusted returns, modifies the rewards when the agent sells, promoting strategies that maximize returns per unit of risk.
- **Penalty for High-Frequency Trading:** To discourage excessive trading, penalties are imposed for frequent buy-sell cycles within short intervals.
- **Buying Penalty:** The agent gets a negative reward for buying based on the trading fee and a penalty that increases with the number of consecutive buy actions, encouraging the agent to diversify its actions and not just buy consecutively.
- **Holding Penalty:** The agent gets a slight negative reward for holding that increases as time goes on in order to help persuade eventual action.
- **Do Nothing Reward:** The agent is slightly rewarded for doing nothing, meaning it has no holdings and is choosing to wait. This will hopefully reward the agent for choosing to avoid unrewarding situations.
- **Expected Reward:** Utilizing the TD-error from the PER, the agent's reward is adjusted based on this expected reward versus the current reward.

Rewards are normalized to a logarithmic scaling to manage the scale of positive and negative rewards.

6 RESULTS

Here we attempt to provide results show the progress of the agent during its first curriculum training. The following results are from training the agent on data from 2016 - 2023 over the course of 50 epochs. The agent is trained on 75% of the data and tested on 25%. This data consists of 'Episodes' representing a single 2-week trading period after a unique Form 4 SEC filing. One epoch is completed when the agent experiences all episodes. When we mention a 'Run' we are referring to a training cycle consisting of 5 to 10 epochs.

6.0.1 ***Agent Action Analysis.*** Throughout the training cycle of the agent we began to recognize a pattern emerging in the agent's actions as depicted in Figure 6, predominantly opting to buy, hold, or do nothing while refusing to sell. This behavior seems to reveal a flaw in the reward mechanism for the sell action. This behavior increases as exploration rate decreases. By Run 7 the agent had trained for 40 epochs.

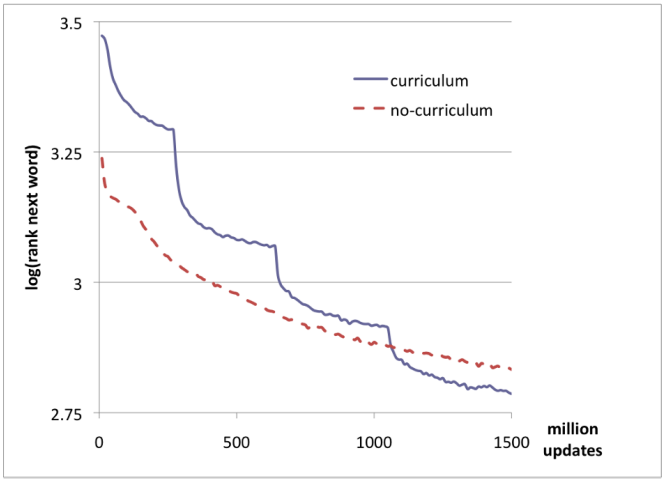


Figure 5: Example language model trained with vs without curriculum on Wikipedia. [4]

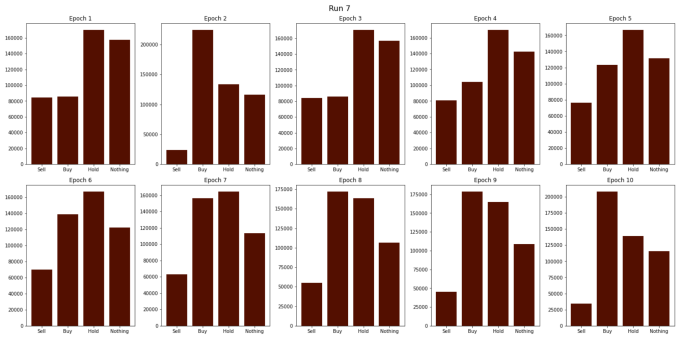


Figure 6: Actions taken by the agent during training Run 7 demonstrating a predominant choice of buying, holding, and doing nothing over selling. As each epoch increases the epsilon value decreases.

6.0.2 ***Training Loss Analysis.*** The training phase of Runs 3-7 showed almost identical mean-squared error (MSE), seen in Figure 7 loss over the training period with each run leading to a slightly lower MSE. This is promising as it shows the agent is learning efficiently as we can see the initial higher MSE values reflect the exploration stage of the agent and as the exploration rate decreases we see a decrease in the MSE.

6.0.3 ***Portfolio Value Analysis.*** The agent is incentivized to maintain a positive portfolio value to maximize realized gains when a sell action occurs. Plotting the agent's portfolio value during testing as seen in Figure 8 shows the fluctuating yet generally increasing trend of the portfolio value across runs. This shows the agent's reward mechanism seems to be on the right path. It must be noted that in the current curriculum, the portfolio value is only positive or negative when the agent is actively holding securities so this metric does not represent the total profit of the agent.

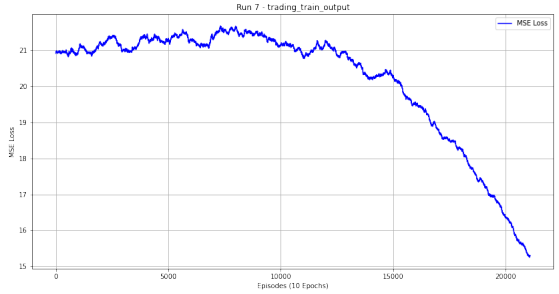


Figure 7: Declining MSE loss over episodes in Run 7, indicative of the agent's improved predictive accuracy through the training progression.

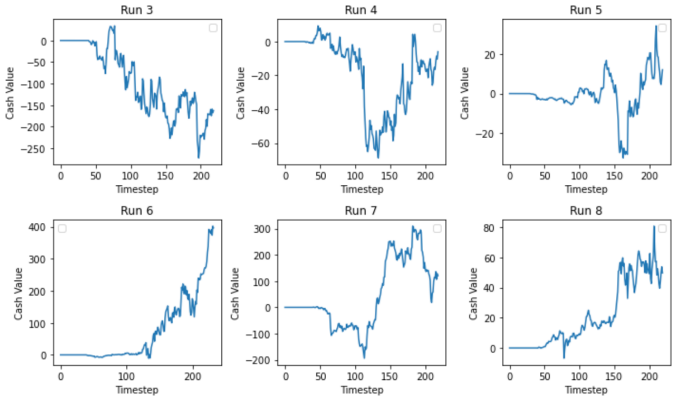


Figure 8: Test results indicating the agent's propensity to build portfolio value across time prior to selling

6.0.4 ***Agent Profit Analysis.*** In order to assess the agent's profit, we take the percent profit from each episode and sum them to get an overall positive or negative net profit. In Figure 9 we see the agent has made 52% profitable trades and 48% unprofitable trades. We can see the large swings in profit which signal that we must implement risk-aversion methods and improve the agent's decision-making strategies.

In order to assess the total overall profit of the agent over the course of all 793 testing episodes we calculate the sum of the percent profit gained or lost at the last timestep of each episode and check if a positive or negative profit was obtained. In Figure 10 we see an impressive total percent profit of 19,156%. This needs to be taken into context though. Although this is an extremely large percentage gain, in order to achieve this you would have to purchase all 793 stocks times the amount of buys the agent performed within each 14-day trading window. Although technically feasible with fractional shares, it would not be advised until the curriculum is improved to emphasize risk aversion and minimize loss. The stark contrast in trade outcomes—significant gains mirrored by substantial losses—raises questions about the model's robustness and risk management efficacy.

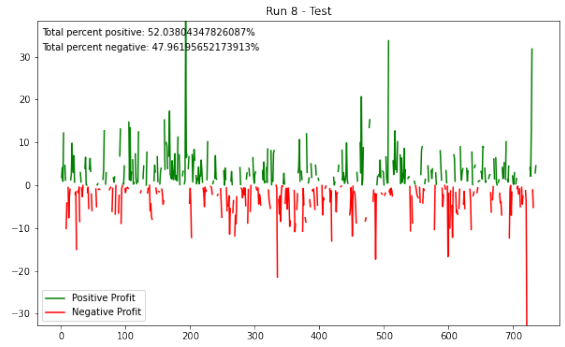


Figure 9: A representation of the distribution of profitable and non-profitable episodes during testing in Run 8. The x-axis represents % profit gained per episode and the y-axis represents the episode

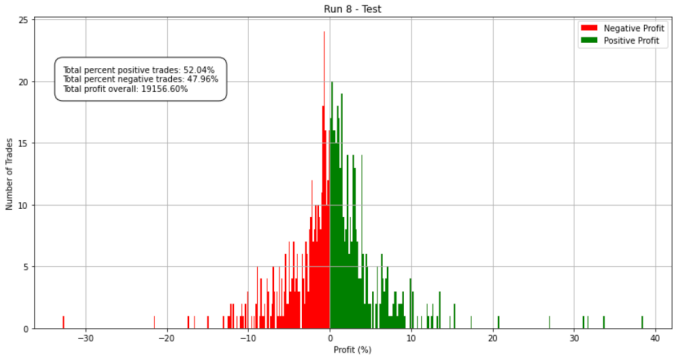


Figure 10: The profit bar chart for Run 8's test phase, demonstrating a significant spread in trade outcomes that signifies the agent's risk exposure.

7 DISCUSSION

The agent presented shows promise after its first curriculum training but there is much to be discussed and many improvements to be made. After accessing the results, many of which were not able to be presented within this paper, we need to address the following concerns in our training 2nd Curriculum:

(1) Profit Variance
(2) Reward Mechanisms
(3) Prioritized Experience Replay Window
(4) Improved Metrics

7.0.1 Addressing Profit Variance. The results, although almost always in favor of profit, were concerning as the variance in profitable vs unprofitable trades bordered on 50% and in many cases the losses would be very large. In order to address this we plan on implementing dynamic stop-loss logic into the action space of the agent. This will help prevent the agent from losing extreme amounts and produce a more conservative trading agent.

7.0.2 Curriculum 2 Reward Updates. Alongside adding a stop-loss mechanism based on the agent's current profit, a few adjustments to the current reward incentives will be performed. In Curriculum 2 we aim to address the issues surrounding the sell action. As seen in Figure 6, the agent was beginning to stop selling as the epsilon value decreased, eventually leading to the agent not selling at all which can be seen in Figure 10 which shows a continuous portfolio value growth. Since the portfolio value is reduced to zero after every sell, the plot shows the agent did not sell throughout each episode. What it does show is that since the portfolio value affects the reward the agent receives, we need to adjust the magnitude at which it affects the reward to ensure the agent wishes to sell. We plan on implementing a depreciating reward as time passes and an increasing reward for selling. Another important update to the reward system in Curriculum 2 involves rewarding the agent based on buys and sells that occur on buy/sell flag signals in the input that we added in our feature engineering step such as MACD crossover, Rate of Change indicator, etc. This should encourage the agent to act on traditionally approved signals. We have noticed that the rewards seem to swing between -30 and 30 and average around -2 to 2 which shows that we need to smooth our rewards to help prevent instability in the learning process. We plan on smoothing and normalizing the reward and then adding any of the extra rewards for actions we promote.

7.0.3 Prioritized Experience Replay. Our current window for the prioritized experience replay was 3 days. We believe this may be too short to obtain the goals we are seeking and are therefore going to read the original paper by DeepMind [13] and determine a better window for our agent to be able to access past experiences from. Increasing this window size will increase the training time so we need to find a reasonable balance. We are contemplating a full 2-week window but since our agent is training on thousands of separate financial windows associated with varying securities, we are unsure as to whether we should allow the agent to maintain its PER memories across episodes or to to clear the memory. More research is needed into this subject.

7.0.4 Improved Metrics. In order to better investigate our agent's behavior, we will be implementing more metrics to help us visualize the progress over time. We added metrics throughout runs during Curriculum 1 which is why our comparative Run to Run results were lackluster. We would like to be able to create a final plot similar to Figure 5.

8 CONCLUSION

Our investigation into utilizing SEC Form 4 insider trading filings through a Dueling Double Deep Q-Network (DDDQN) has unveiled promising yet preliminary results. The agent, designed to capitalize on post-disclosure market adjustments, demonstrates the potential to offer strategic advantages in trading. However, it also reveals significant challenges yet to be addressed.

Moving forward, we aim to refine our trading agent through continuous curriculum-based training leading the agent to learn to navigate real-world trading dynamics. The ultimate goal is to achieve a level of robustness and reliability that allows full integration with the SEC's Real Simple Streams for live data acquisition alongside a trading platform for complete automation. This continuous improvement will be vital in transitioning from yet another theoretical model to a practical trading utility, bridging the gap between academic research and financial technology application.

9 CONTRIBUTIONS

Name	Contributions
Eric B. Martin	- Data acquisition - Apache Spark code development - Setting up the framework and preparing to start the model - Model development and training - Hyper-parameter tuning - Paper write up - Considering future goals of implementing real-time analysis and actions
Tarun Sai Pamulapati	- Initial data preprocessing - Exploratory data analysis - Apache Spark code development - Feature engineering - Setting up framework and preparing to start model - Analysis of results - Paper write up - Considering future goals of implementing real-time analysis and actions

ACKNOWLEDGMENTS

Thanks to Dr. Sangmi Lee Pallickara for providing us with more memory in order to run on our student computing environments and also for aiding in the acquisition of our high-rate Alpha Vantage API of whom we thank as well for granting us access to their free high-speed academic license. Thanks to David Gallo as well for inspiring the use of a DDDQN through his excellent thesis at the Toulouse Business School. [7]

REFERENCES

[1] [n. d.]. Rate of Change (ROC) [ChartSchool]. https://school.stockcharts.com/doku.php?id=technical_indicators:rate_of_change_roc_and_momentum. Last accessed 24-APR-2024.
[2] [n. d.]. Volume Weighted Average Price (VWAP) [ChartSchool]. https://www.investopedia.com/terms/e/ema.asp. Last accessed 24-APR-2024.
[3] Alpha Vantage Inc. [n. d.]. *Alpha Vantage.* Alpha Vantage Inc. Available at: https://www.alphavantage.co/. Last accessed 13-MAR-2024.
[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/1553374.1553380
[5] James Chen. [n. d.]. What Is EMA? How to Use Exponential Moving Average with Formula. https://school.stockcharts.com/doku.php?id=technical_indicators:vwap_intraday. Last accessed 24-APR-2024.
[6] Jason Fernando. 2019. Moving Average Convergence Divergence – MACD Definition. https://www.investopedia.com/terms/m/macd.asp. Last accessed 24-APR-2024.
[7] David Gallo and Neha Chaudhuri. 2022. Algorithmic Cryptocurrency Trading Using Sentiment Analysis and Dueling Double Deep Q-Networks. https://www.monkeydg.com/assets/docs/David_Gallo_MSc_Thesis.pdf.
[8] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. (2010). http://yann.lecun.com/exdb/mnist/
[9] Cory Mitchell. 2019. Using Bollinger Bands to Gauge Trends. https://www.investopedia.com/trading/using-bollinger-bands-to-gauge-trends/. Last accessed 24-APR-2024.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533. https://doi.org/10.1038/nature14236

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[12] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf

[13] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. Prioritized Experience Replay. arXiv:1511.05952 [cs.LG]

[14] William F. Sharpe. 1998. The Sharpe Ratio. *Streetwise – the Best of the Journal of Portfolio Management* 3 (1998), 169–185.

[15] United States Securities and Exchange Commission [n. d.]. *United States Securities and Exchange Commission.* United States Securities and Exchange Commission. Available at: https://www.sec.gov/. Last accessed 09-MAR-2024.

[16] United States Securities and Exchange Commission [n. d.]. *United States Securities and Exchange Commission Public Datasets – Form 3,4,5 Filings.* United States Securities and Exchange Commission. Available at: https://www.sec.gov/dera/data/form-345. Last accessed 09-MAR-2024.

[17] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. 2016. Dueling Network Architectures for Deep Reinforcement Learning. arXiv:1511.06581 [cs.LG]

[18] Zhengwei Zhu, Can Hu, Chenyang Zhu, Yanping Zhu, and Yu Sheng. 2021. An Improved Dueling Deep Double-Q Network Based on Prioritized Experience Replay for Path Planning of Unmanned Surface Vehicles. *Journal of Marine Science and Engineering* 9, 11 (2021). https://doi.org/10.3390/jmse9111267