Investigation into a Practical Application of Reinforcement Learning for the Stock Market

Philip Traxler  
*Southern Methodist University*, philipltraxler@gmail.com

Sadik Aman  
*Southern Methodist University*, saman@smu.edu

Will Rogers  
*Southern Methodist University*, willrogers@smu.edu

Allyn Okun  
ajokun@gmail.com

Follow this and additional works at: [https://scholar.smu.edu/datasciencereview](https://scholar.smu.edu/datasciencereview)

Part of the Data Science Commons, and the Finance and Financial Management Commons

**Recommended Citation**

Traxler, Philip; Aman, Sadik; Rogers, Will; and Okun, Allyn (2023) "Investigation into a Practical Application of Reinforcement Learning for the Stock Market," *SMU Data Science Review*: Vol. 7: No. 3, Article 6. Available at: [https://scholar.smu.edu/datasciencereview/vol7/iss3/6](https://scholar.smu.edu/datasciencereview/vol7/iss3/6)

This Article is brought to you for free and open access by SMU Scholar. It has been accepted for inclusion in SMU Data Science Review by an authorized administrator of SMU Scholar. For more information, please visit [http://digitalrepository.smu.edu](http://digitalrepository.smu.edu).
Investigation into a Practical Application of Reinforcement Learning for the Stock Market

Sadik Aman\textsuperscript{1}, Will Rogers\textsuperscript{1}, Philip Traxler\textsuperscript{1}, Allyn Okun
\textsuperscript{1} Master of Science in Data Science, Southern Methodist University, Dallas, TX 75275 USA
\{saman, willrogers, ptraxler\}@smu.edu

Abstract. A major problem of the financial industry is the ability to adapt their trading strategies at the same rate the market evolves. This paper proposes a solution using existing Reinforcement Learning libraries to help find new strategies at a practical scale. Using a wide domain of ticker symbols, an algorithm is trained in an environment that better represents reality. The supplied decision-making algorithm is tested using recorded data from the U.S stock market from 2000 through 2022. The results of this research show that existing techniques are statistically better than making decisions at random. With this result, this research shows how a practical application of Reinforcement Learning is possible through the inclusion of many more ticker symbols than previous research has done before. However, there is still work to be done to achieve acceptable returns. Potential applications of this research include informing human traders or creating automated traders.

1 Introduction

The stock market is undeniably the backbone of the American economy and one of the pillars of the world economy. Whether it is through retirement accounts, livelihoods, or even political effects, the transactions created in the market influence every person. Human traders create tens of billions of dollars in transactions every single day in the United States. Each transaction, colloquially known as stock trading, is the process of buying and selling stocks. This is done to generate profits and wealth. A trader is a person who participates in stock trading, either as a career or hobby. The typical skill set of a trader includes information analysis, market domain knowledge, and mathematics. Additionally, the best traders will have an intuition for price fluctuations over a period. Traders will leverage the knowledge they have learned of trends to complement their book learning. Combined with their assessment of any risks involved, these traders make buying and selling decisions. The other major types of trading include day trading, scalping, swing trading, and position trading. These other types of trading follow a similar operating pattern, but the rules, techniques, and intuition of each vary. Besides following a similar pattern for doing their job, good traders all share another, less quantitative quality. Namely that they will continue to learn from their respective environments throughout their career and will receive positive and negative reinforcement for all trading actions taken. These lessons are then used to create trading strategies that can recreate success while avoiding past failures.

Profitable stock trading strategies are vital to investment companies and hedge funds. Those entities generate their business from having the best strategies that
generate the most revenue for their customers. Since the market is consistently evolving, new strategies are often needed to best operate in an unfamiliar environment. Recent technology, new companies, and new culture are just some of the ways that render old strategies obsolete. Strategies are engineered to provide an edge over other traders and strategies to increase profitability. These strategies are developed by experienced traders with the goal of optimizing returns. The optimization parameters balanced are fundamentally the potential return on investment and level of risk with that investment. Although creating a strategy is simple in theory, the application of a dynamic and profitable strategy is often the most difficult to implement. Every company/trader search for the best strategy and those that develop a winning strategy first get the best rewards. Thus, investment firms spend considerable effort hiring the best traders in the name of maximizing profits.

However, traders, by nature of being human, take a long time, potentially a lifetime, to learn all the rules necessary to become consistently effective. Specialized education teaches factors like price patterns, candlesticks, indicators, support, and resistance. A trader's education accelerates the learning process, but still takes many years to develop into a fully mature trader. Complicating matters further is the innate complexity and dynamic properties of the stock market that necessitate periodic strategy updates. One of the key challenges, as seen by the researchers, facing the financial industry today is the ability to dynamically change at the same speed at which the market moves. One solution to this problem is to increase the speed at which traders can learn, to match the constant transformation of the market. To meet this challenge, it is possible to combine the speed of computers with the learning paradigm of humans through Reinforcement Learning.

The method behind Reinforcement Learning is to have an agent interact with a simulated environment, receive rewards for desired behavior, and potentially be punished for misbehavior. The agent's action and observation space and any reward functions are determined beforehand. Through trial and error and continued interaction with the environment the agent learns to maximize its reward. The challenges of this algorithm are accurately modeling the environment used during training and developing the rules to determine the action taken by the agent. The major advantage of this type of learning is that approximate solutions can be found to problems where a closed form solution is impractical. Stock trading is a perfect example of a problem that does not have a single or closed-form solution. When considering the stock market, the appropriate reward to use for the agent can be found in the market itself. Namely total returns over a period. If we associate the utility function with earnings, the stock market seems perfect for applying Reinforcement Learning models.

Previous research applying Reinforcement Learning (Brim & Flann, 2022; Carta, Corriga, Ferreira, Podda, & Recupero, 2021; Kabbani & Duman, 2022; Li, Ni, & Chang, 2020; Lussange, Lazarevich, Bourgeois-Gironde, Palminteri, & Gutkin, 2021; Zhang & Lei, 2022) to the stock market has found that the technique can provide valuable insight. The authors have observed algorithms developed using this method can achieve a sufficient level of performance. Additionally, the authors observed that several different techniques can be used in conjunction with Reinforcement Learning to varying effectiveness. The key difference between the different projects is the way that rules for making a buy/sell/hold decision are created. Despite the differences
between projects, almost all believe that there is still untapped potential applying Reinforcement Learning to the stock market.

A general trend noticed by the authors was a general limit of scope to a small set of ticker symbols. However, this does not provide an accurate model of the environment traders frequently find themselves a part of. While capable of providing useful insight, this limits the practical usefulness of models developed in this manner. The purpose of trading algorithms is to improve upon human intuition through greater processing speed and increased data consumption. By limiting the number of ticker symbols, previously proposed algorithms are specialized to the point that they do not have a practical use. Similar to the difference between high performance race cars and the average daily driver. Arguably, this specialization creates algorithms that would have a similar level of performance as human experts, albeit at a quicker rate. However, through their training across a broad set of tickers, human experts develop a more generalized set of rules for effective trading.

The research presented here proposes to incorporate a wider training domain to Reinforcement Learning. By exposing the algorithm to more training data, it will better emulate the learning process of a human trader. Those same human traders that develop the strategies the affect the lives of each person. The difference being that the training time will be much reduced and more encompassing. Focusing on short term application to the stock market, actions and rewards have been developed towards this goal. Human traders can take the outputs of these models, now capable of adapting to the ever-changing market conditions like a human, to help create trading strategies and inform live-trading decisions.

2 Literature Review

This focus of this research is predicting stock market behavior, and thus the literature reviewed in its preparation, is in the following areas as applied to the stock market: Reinforcement Learning, Feature Extraction, Sentiment Analysis, and Ensemble Models.

2.1 Reinforcement Learning

Applying deep reinforcement learning to the stock market is a recent trend in Data Science. This paper intends to expand upon those ideas. The amount of research in this area has ballooned in recent years and provides a plethora of knowledge and lessons (Kabbani & Duman, 2022). Overall findings suggest that deep reinforcement learning applied to stock market data is overall an improvement over other methods used in the past. One study (Li, Ni, & Chang, 2020) helped to back up this statement by comparing reinforcement learning models to a known algorithm, Adaboost. They found that for experts, the Adaboost algorithm will perform better than the reinforcement learning models they tested against. However, for less familiar stocks, the reinforcement learning models outperformed other methods.

A key aspect of the stock market is reacting to the actions of other players in the market. How well this behavior is modeled will dictate how well any model developed would perform in the real market, outside of the data used to initially train, test, and verify the performance. A new methodology proposed (Lussange, Lazarevich,
Bourgeois-Gironde, Palminteri, & Gutkin, 2021) for modeling certain market behaviors that have been missing from models of the past. After developing their new model, based on a MAS architecture, they found that they were able to accurately model these missing behaviors, using the London Stock Exchange and found their model “emulates key market statistics of real stock exchanges”.

Several papers brought some novel concepts when applying reinforcement learning to the stock market. Developing a two-pronged approach (Ma, Zhang, Liu, Ji, & Gao, 2021), some research divided the problem into two components, one layer to learn and predict from the current state of the market and a second layer to account for long-term historical trends. This appeared to lead to improved performance over traditional methods for their limited scope. Another paper (Serrano, 2022) attempted to find gains in performance using random neural networks while including previous learnings and not just the current state of the model. The research led to marginal performance increases at an increased computational cost for an overall mixed outcome. The background on reinforcement learning leading up to their research was a noteworthy aspect of the research. Others (Zhang & Lei, 2022) incorporated policy gradient with reinforcement learning. Improvements in performance were found when compared against traditional and textbook base methods. Improvements to their research would be to increase the forecast frequency to the minute scale, something necessary for short-term trading, as well as better accounting for volume. Additional research (Théate & Ernst, 2021) researched focused on maximizing the Sharpe ratio performance indicator using reinforcement learning. This research used a new algorithm based on the Deep Q-Network algorithm which was then altered to better fit the application of stock market trading. The conclusion of the paper was that, although promising, the new algorithm did not perform as well as other strategies that target specific market environments.

A limitation of about every study examined was the exclusion of sentiment analysis and behavioral aspects of trading. Additionally, studies in this area generally limited their scope to a single stock and not a portfolio. Expert traders are expected to use both in their decision-making process as they help to expand the number of opportunities to maximize the performance of investments.

2.2 Feature Extraction

Another major aspect of the literature reviewed for this research was in the category of feature extraction. These features are to be used in the reinforcement learning models to maximize their scoring functions and create the desirable outcome of maximizing performance. Features can take many different forms, so several distinct types of features and how to extract them were examined to build up a library of techniques to leverage.

One study started by acknowledging that the modern trading environment is global, which means the different markets can influence each other. Another way of stating this is that the changes that happen during the overnight period of one market have an effect the next day of another market. Using a method proposed (Gao, Zhang, Zhang, Zhao, & Wang, 2022) in the research called Multiple Branch Convolutional Neural Network based on Genetic Algorithm, the researchers were able to create a model with features that incorporated the overnight influence of international markets on domestic
ones. Their proposed model and features were found to have improved accuracy over competing methods using a seven-year span of historical data for 30 SMIs.

Another research project (Htun, Biehl, & Petkov, 2023) focused on performing a survey of other research targeting feature extraction techniques to compare their respective uses for any trends that may appear. The article goes on to explain the different methods used for stock market prediction and the importance of feature extraction as applied to stock market data. Using a keyword search on a specified date, the 32 relevant papers were pulled for use in the study. The feature extraction techniques found across all the papers and generates a table to use for quick comparison of all the techniques and background of their use. Concluding findings included a lack of ensemble models and either basic or technical features applied to generated models, rarely both at the same time.

There is evidence that existing methods for evaluating the stock market do not work well in all market conditions (Wu, et al., 2020). They argue that by applying a Gated Recurring unit, informative financial features can be extracted. When combined with known deep reinforcement learning methods, single stock trading performance is improved and stabilized vs other strategies. Verification of results was done using stock market data from multiple countries. The proposed improvements to existing models showed consistent returns over the reference turtle method for most cases examined.

A novel neural model is proposed (Taghian, Asadi, & Safabakhsh, 2022) to help extract features from data and continue to learn new ones as the model. The trading model is subjected to certain requirements by the authors. Namely, dynamic environments, feature extraction, and adaptability to the market the stock exists inside. The generated methods depend on candlestick charts, finding certain patterns to generate trading rules. Once developed the performance against typical baselines methods for a handful of selected stock data was used to measure performance. Results showed that the asset-specific method proved more profitable than general rules. In addition, the asset specific method proved more robust to differences between training and real data.

The overall limitations of research done in this area were the intuitiveness and applicability of the features extracted.

### 2.3 Sentiment Analysis

Experts in the field have long assumed, and try to leverage the fact, that the overall sentiment around a stock and the market has a relationship with their respective behavior. The general statement that when people feel good about something leads to better performance makes logical sense, but the exact relationship is the subject of several research projects. One of these (Liapis, Karanikola, & Kotsiantis, 2023), created a process to examine several features and algorithms across a variety of datasets to check how the inclusion of sentiment improves models. Creating a sentiment time-series for each stock under examination, evidence was found by the researchers that suggests the inclusion of sentiment when predicting stocks is generally beneficial. However, they noted that exact performance depends on the algorithm used for sentiment extraction and the method of forecasting.

Some research in the area (Chen, Zhao, Li, & Lu, 2020) focused on getting a better understanding of the relationship and testing one of the assumptions. Namely, that the sentiment of investors leads to actual volatility. The research created proxy variables
were created to help measure the sentiment of the market, which were then used to test for a lead-lag relationship. When the researchers tested market data from China, it was found that there is evidence to support a claim on a relationship between investor sentiment and market volatility with volatility lagging sentiment by about a month.

Other researchers (Cristescu, Nerisanu, Mara, & Oprea, 2022) point out how system crisis can make existing models for evaluating portfolio risk low performing. Due to the nature of the stock market, investors often use their sentiment to make decisions. The proposed solution is to incorporate features based on sentiment to see if models fit data better. Using regression-based techniques, incorporating sentiment as an exogenous variable, they found that the R coefficient could be raised by a measurable amount when performing regression. In additional research (Koratamaddi, Wadhwani, Gupta, & Sanjeevi, 2021), sentiment was partnered with Reinforcement Learning to examine more than just the quantitative data when developing and testing models. Another interesting example of incorporating sentiment analysis to predicting stocks prices (Lima Paiva, Felizardo, Bianchi, & Costa, 2021) examined the integration of Reinforcement Learning and Natural Language Processing. The product of their research was a custom algorithm that took advantage of market sentiment. Using this “trading system”, the researchers were able to deliver performance in traditionally difficult to predict market conditions.

2.4 Ensemble Models

Often, the best performing models are created by combining different models to create something better than the sum of their parts. However, it is also possible to create something that is less than the sum of its parts. Finding the right balance of different models has thus been the subject of different research (Yang, Liu, Zhong, & Walid, 2020). When it comes to the stock market, several types of models may be appropriate, so the number of combinations is correspondingly high. One group of researchers (M. Ameen Suhail, et al., 2021) studied how different types of analysis, technical analysis, time series, and fundamental analysis affect a stock’s price and direction. Traditionally time series models handle stationary data well, but struggle with a live trading scenario. Including seasonality and other terms helps to improve the model but leaves room for improvement still. Their research found improvements could be made by combining deep learning neural networks with multivariate time series to create a well performing ensemble model.

Other researchers (Carta, Corriga, Ferreira, Podda, & Recupero, 2021) critiqued the work of other researchers by suggesting that despite finding success through ensemble models, they are leaving performance by not optimizing. Instead of combining models at random, their research took models with low correlation so that they could be used together in a more optimal fashion. Specifically related to this research, this method was focused on intra-day trading and used real stock data from JPMorgan and Microsoft to measure their performance. In a related topic research was done (Yu, Wu, Liao, & Han, 2023) focusing on developing multiple models that are suited for different aspects and times of the market. Instead of trying to build a single model that could handle all cases, their research built a method to weigh and select from a list of models to use in combination. Targeting an entire profile, they were able to create what could be classified as an ensemble model that generated better return with risk when validated with real market data.
3 Methods

3.1 Key Techniques

Reinforcement Learning

Two well known branches of machine learning are supervised and unsupervised learning. Many of the algorithms that everyday people associate from machine learning are from one of these two areas. Reinforcement learning is another branch of machine learning that focuses on using agents that make decisions in each environment to maximize their reward.

The key aspect of reinforcement learning that makes it different from either supervised or unsupervised learning is that the goal is more important than the technique. The main guiding factor for reinforcement learning is a utility function, also known as a reward function, that the agent in the environment is trying to maximize. The general operating procedure for a reinforcement learning model is to have a set of states for the environment and the agent(s). Each agent has a set of actions it can take influenced by those states. Each action gets a corresponding expected reward based on the following state of the agent after that action. Each agent goes through this process independently and each decides on the action it will take. These actions are then taken and communicated to the environment, which then updates its own state and corresponding rewards are then distributed to the agents. The entire cycle is then repeated until the process is determined done. The design of the process is for the agent to learn optimal or near-optimal behavior that will maximize its reward function during training. Depending on the use case, the behavior can either be frozen in place or allowed to continue to develop, when deployed outside of the training environment.

As with any algorithm, Reinforcement Learning has certain use-cases where it is appropriate, needs in order to operate, and general advantages and disadvantages. It can be argued that the biggest advantage of Reinforcement Learning is the inherit approximation of solutions that occurs. In other words, the solution found will be an approximation of any actual solutions that exist. Therefore, Reinforcement Learning is more appropriate for problem spaces that either have prohibitively large or non-existent exact solutions. The other big use-case for Reinforcement Learning occurs on the opposite side of the equation when the agents operate inside a complex environment, or the actions of the agent itself are intertwined with the environment. Once it has been determined that Reinforcement Learning is appropriate, the algorithm has requirements in order function. The first of these is an observable environment or observation space. This can take many forms, but will be most recognizable as either a simulation, recorded data, or a real environment. The second requirement for the algorithm is a set of actions for the agents to select, referred to as the action space. The creation of the actions and how to select them is where the major difference between different application of Reinforcement Learning is located. With these basic needs, the algorithm can be applied to the problem being worked. The advantages that come from applying Reinforcement Learning to the problem being worked include the approximate solution, modeling complex interactions, and the similarity to the way many organisms in nature operate. Conversely, the disadvantages of Reinforcement Learning includes the
need for a well-modeled environment, the lack of a guarantee for an optimal solution, and the need to have a well-defined reward system.

As this research is paying special attention to the stock market, additional considerations are required to effectively apply reinforcement learning. Among special considerations required is an understanding of how well the stock market is suited for reinforcement learning, given earnings over a period of time serves as the reward mechanism, currently no exact solutions exist, and the environment is extremely complex. Furthermore, reinforcement learning can be given the ability to balance long and short term rewards. Another hurdle to using market data and market conditions as the base for reinforcement learning is as the number of actual variables is enormous and continues to grow. Feature engineering or feature extraction is almost a necessity to accompany the development of rules for the agents.

**Recurrent Neural Network**

A Recurrent Neural Network (RNN) is a specific type of neural network designed to work well for time-series-based data. A simple diagram of how a RNN works can be seen below in Figure 1. Here, the RNN has been unrolled to show how it is working. In practice there is only a single layer that gets re-used for each step in the data. The general premise is to use the outputs of the previous step in the model as a set of additional inputs into the next step. Each training step is then sent through the model sequentially, as opposed to batched input like a nominal neural network. Once all training sequences are sent through the network, the weights of the model are updated. This process is repeated until a stop condition is met.

![Unrolled Recurrent Neural Network architecture](https://scholar.smu.edu/datasciencereview/vol7/iss3/6)

**Figure 1 : Unrolled Recurrent Neral Network architecture**

The Long Short-Term Memory (LSTM) model is a derivative type of RNN from the base described above. One of the shortcomings of a traditional RNN is that it only has roughly five or six-time steps worth of memory. This is due to the vanishing gradient effect of neural networks. The LSTM solves this problem by using a series of gates to
determine which information should be forgotten, which information should be added into, or combined, and which information should be used, in the current memory state.

Stock data is inherently time-series data since each step is correlated to previous steps of the market. However, the relationships are complex enough and have sufficient structure on different time scales that 6 steps of memory are not expected to provide robust predictive power. The LSTM can extend the amount of memory available to the model which will enable the model to exploit relationship with more previous time steps. As the data for this research is quarterly, this means that multi-year seasonal trends will have several examples to learn from instead of just a single instance that might have occurred in the last six quarters.

3.2 Data

Source

The sources for our data is rooted in SEC filing created by each company during their quarterly and annual earnings calls. From within the SEC filings, the income statement, balance sheet, and statement of cash flows are delineated and stored for later use within the overall algorithm. Although the information is derived from SEC filings at the time of the earnings call, one of the limitations these filings currently face is the ability for the company to amend the documents after they’ve been published. It is in this fact that the research could be limited to the initial data found.

Data Sources:

a. The data will come from SEC filings in the form of 10-K, 10-Q, and 8-K. These are standard filings required by a publicly traded company.

b. Simfin.com

c. Via the documents, we can extract the income statement, balance sheet, and statement of cash flows.

d. Dataset including sentiment data extracted from standard filings to be used in reinforcing the fundamental analysis.

The actual means of extracting fundamental financial data for use in the research is accomplished through the SimFin API. They publicly show support for over 620K financial statements, 7000 metrics per day, 5000 stocks, with over 20 years of historical data. All their data is verified, and any user can check it for themselves if they want. For this research, this means that the data used is guaranteed to be accurate and accessible. This will add the benefit to this research of being easily repeatable without direct access to a copy of the exact data files used for this research. This was observed as a noted weakness of other research in this area.

Given the size of the data available through SimFin, a subset is required for the purposes of this research given computing constraints, limit the scope of this research, and allow timely exploration. One limitation applied for this research is only using US companies. Also, the time step of each data point is quarterly from 2000 through 2022. Variables for each ticker in the data set are captured in Table 1 of Appendix. As a summary, there are 67 unique variables in the dataset to use as the environment for testing the Reinforcement Learning agents. The list of ticker symbols included in the original dataset can be found in Error! Reference source not found. of Appendix. The tickers as part of this subset were then used to lookup the closing price from the
day the report was published as well as 10 days after the report was published. The price after the publish date was not a part of the input into the model but used by the environment to distribute rewards.

**Exploratory Data Analysis**

The exploration of the dataset from SimFin illustrates correlations between Cash, Cash Equivalents, & Short-Term investments and Total Shares. There is an additional strong correlation between Total Shares and Total Current Assets. From this point, it is clear the Total Shares will be a variable that is paid extra attention to further into the analysis. The results of the correlation study can be found in Figure 2.

![Correlation coefficients of variables in dataset](image)

**Figure 2:** Correlation coefficients of variables in dataset

At first glance, missing data appears to be a major problem with this data set, as seen in Figure 3. However, these missing values are simply unreported on non-applicable fields to that ticker symbol. In other words, they can be assumed to be zero. Following this methodology, missing values in this dataset were filled with zeros. Plots showing the missing values for the remaining features of the data set can be found in Figure 11 and Figure 12 in Appendix.
Several different sectors are present in this data set as seen in Figure 4. Through discussion with the algorithm investing SME available, it was determined that it would be advantageous to have different models for each sector. For the purposes of this research, it was determined that focusing on the technology sector would be sufficient to build a proof-of-concept model that could later be retrained on data from the other sectors.

Figure 3: Features in data with most missing values

Figure 4: Composition of data by sector
3.3 Proposed Solution

The solution proposed is to combine the fundamental analysis from the delineated SEC filings along with the sentiment of the filing itself to better understand if the company is maintaining a buy, sell, or hold trade. The changes within the balance sheet, income statement, and statement of cash flows from quarter to quarter and year to year illustrate a larger image in the perceived values of the company. Additionally, the sentiment gathered from the filing may contradict or further confirm the results of the fundamental analysis.

For the Reinforcement Learning agents, the proposed algorithm is going to use a Recurrent PPO Model from the SB3 package with the “MlpLSTMPolicy” option. Extract features from the past history of all stocks under consideration. The observation space available to the model will include all of the fundamental data from quarterly reporting as well as pricing data from the day the report was released. In addition to this information for all stocks available during the time period of the dataset, the agent will also have access to its own state of cash and holdings. All of this information is then flattened into a single observation vector that the model will be fed at each timestep. With this information, the model will have an action space composed of a vector the length of all tickers available in the dataset that can take a value of buy/sell/hold. The assumed action is hold/no-action for all values of the vector, including any stocks that may be invalid at a particular time step.

The decided action for each ticker symbol will be taken on exactly one share of that stock. The model is not allowed the freedom to choose the amount of shares for each stock in order to limit the decision space and help the model find a solution in a reasonable amount of training time. An effect of this decision on performance is that the model will be biased towards stocks with larger prices since, smaller percentage changes there will lead to bigger returns than larger changes on a stock with a smaller price.

3.4 Evaluation Methodology

In order to test the algorithm, a simulation of the stock markets is needed. The data gathered for this research will be treated similar to a traditional time series dataset. The first part of the data will be used to create an initial agent. The series will then be stepped through, where the agent will make buy/sell/hold decisions. The simulation will then step forward and rewards given to the agent based on the recorded market data. Since the data is already recorded, the actions taken by the agent are unable to be factored into the response of the environment. If we are able to assume that the actions of the agent are not enough to majorly influence the market, then this is not a major issue. Therefore the evaluation of any given simulation will be limited to a single agent.

As a control group for this research, an agent will be introduced that makes decisions at random. This to provide a reference for any improvements that are from the algorithm performance vs any trends that might have been present in the market. In an upward trending market, random decisions should still end up with positive returns since just having money in the market was a good choice. Any algorithm that performs worse than this is effectively losing money. With this agent acting as a normalization factor, any performance gains by the agent will be able to be fairly assessed.
Once both models have completed their training, the result of the test set at each time step for both the proposed algorithm and the baseline reference algorithm will be differenced. A positive difference meaning that the proposed algorithm performed better at that time step. Treating this difference as a sample a simple t-test will be performed to see if the difference is statistically greater than zero. If the proposed model passes this t-test, it will mean that there is enough evidence to support the claim the proposed model is better than the baseline.

4 Results

4.1 Study Results

The purpose of this research was to create a practical application of Reinforcement Learning to the stock market that can supplement the capabilities of short-term traders.

The model was given 100 total runs through the training data to learn the policy necessary to successfully navigate the stock data. As a reminder, this entailed 84 discrete time steps over Y2000Q1 through Y2020Q4. In Figure 5, the value of each training iteration over is plotted. The starting cash given to each iteration was $10,000, where it was allowed to make decisions as it decided.

![Figure 5: Portfolio value over training period](image)

Once the model was trained, the rest of the data was used to evaluate how well it could handle new data. This split represented a 90/10 train/test split. A similar plot to the one above for the testing period can be seen below in Figure 6. As opposed to the training data, the test data is run through 10 times.
Figure 6: Testing results of trained model

As a fair comparison, the test data was used in a competing model that made decisions at complete random. The same approach as with the proposed model was used and the resulting performance for "training" can be seen below in Figure 7. In addition, the test data received the same treatment, and the results are in Figure 8.

Figure 7: "Training" results for random actions
The random model’s purpose was to serve as a baseline to compare performance against the proposed model and account for any inherent trends in the data. As a starting point, the difference between the value of the proposed model and random model was done for every time step. In Figure 9 below the result of this operation is plotted. The dark blue line represents the average difference (value above baseline) at each timestep for the proposed model with the shaded area representing the intervals of the different runs through the test data. The further above zero these two entities are, the better the proposed model is performing.
The difference data was treated as a single sample set which can be seen above in Figure 10. Testing if this sample was statistically greater than zero resulted in a p-value of 0.005. At an alpha level of 5%, there is enough evidence to support a claim that applying Reinforcement Learning to a broad spectrum of tickers results in a model that is statistically different from taking actions at random. There is evidence to support our hypothesis of Reinforcement Learning having a practical stock market application.

5 Discussion

5.1 Related Results

The algorithms used in the research here are either established techniques or leveraged from previous research. Instead of looking for new and novel ways of approaching an important problem, this research has chosen to focus on maturing existing technologies. As a result, there are many similarities between this research and those referenced. This is overall consistent with other research in the field.

Several studies described in section 2.1 showed that satisfactory results were achievable when limiting the scope of the model to a handful of tickers. Displaying similar behavior, this research has shown that lessons from smaller scale projects are potentially applicable in a more practical setting, once a few obstacles are overcome.

The defining feature of this research was widening the amount of ticker symbols allowed to the agent when making decisions. The major advantage of this is that it better represents the trade space available to human stock traders. Doing so, this research has shown that it is possible for Reinforcement Learning to be applied to the stock market in a practical situation.

From an overall viewpoint, while the results of this research are consistent with precious research, they are not well related.
5.2 Findings

This research started with the main question of if it was possible to emulate the behavior of an experienced trader using readily available algorithms and technology. The direct answer to this question is that there is evidence to support this claim, but there is still work required to reach this point. The LTSM-based model showed it could learn, even with a large decision space to explore and a limited amount of training runs. This is most evident in Figure 5 where the training results were presented.

However, this success did not translate to the testing data seen in Figure 6. The implication of the poor performance on the testing data is that the model presented here is not ready for use in a real trading situation. Hypothesized reasons for the poor testing performance are headed by either a lack of sufficient training cycles or a discrepancy between the format of the training data and testing data. For instance, if the test data did not have the correct number of time steps included before attempting to extract a prediction.

One lesson learned during this research included tuning the reward function to encourage profit-making behaviors while discouraging the model from exploitative behavior. For example, in the early versions there was a tendency for the model to take lots of invalid actions because the environment would just ignore these anyway. While performance seemed good, it created an overly aggressive model that performed extremely poorly during testing. This was mitigated by including a penalty term for invalid actions, but not eliminated.

5.3 Limitations

A major limitation of this research was using recorded stock market data as the environment the agent operated in. This did not allow the agent to affect the environment and thus forced the agent to remain small enough that its effect could be declared negligible. Additionally, the model was not allowed to decide on the number of shares that any decision was made on. The value for the number of shares was always equal to one. It is also worth restating that the results presented from this research are only applicable to the technology sector.

The biggest limitation of this research is the number of computational resources available for training the model. Reinforcement Learning general guidelines suggest at least several hundred cycles through the training data. The combination of using a RNN with a relatively large data set applied to RL created a model that required more computing time than was reasonable on the hardware available to researchers.

5.4 Future Research

While not fully mature, there is enough evidence from this study and others like it to suggest that there is a future for reinforcement learning applied to the stock market. Specific extensions of this research would include further refinements of the reward function used in the model, building of a custom model, and reducing some of the restrictions placed by the environment to reduce the solution space.

It was found during the study that our specific model was sensitive to the reward function. While the sensitivity was expected, the degree of this sensitivity was surprising. Looking for a more stable reward function would benefit the training of the model and potentially create a more extendable model.
Since the goal of this research was focused on testing the readiness of an existing model to reach a more practical scale, a custom model and policy were not considered as in similar research. However, it became clear that the model's performance could be increased if some custom steps were included. For example, instead of allowing the model to take positions anyway it saw fit, there could be a step that focuses resources on the top predictions. This was seen as a desirable behavior but could not be implemented without creating a custom model and policy.

Related to the custom model is the restriction placed on it by the simulated environment. For instance, this research used a playback method for market simulation. It would be interesting to see if the model performed similarly when actions taken influenced the next simulation step. Also, the model's observation space was limited to observations of the market itself. Allowing the model to see the actions of other actors in the market, as some investors do, may help to increase the robustness of the model. Finally, there are several separate ways to interact with the market beyond buy, hold, and sell that were not covered under the scope of this research.

5.5 Ethics

As with many artificial intelligence projects, the ethical questions of whether this should be done or not are asked. This project has limited its scope market capitalization and prediction. However, Reinforcement Learning has continued to prove its ability to adapt to a wide variety of problems. While not the intent of this project we have shown evidence that the algorithm can handle problems that were once seen as impossible for machines due to their complexity and size. Yet, the solutions found are not necessarily constrained by ethics. Let this be a reminder to all consumers of this research that Reinforcement Learning is designed to maximize its reward only with no consideration as to the implications of its actions.

6 Conclusion

The implementation of reinforcement learning in stock trading is at the initial and growing stage of the financial market. From the research, we found that when reinforcement learning is applied, it improved the model's performance in predicting the stock's accuracy and profitability. We have also found that reinforcement learning performance depends on the price pattern, duration of the stock yearly, quarterly, short-term and long-term. From the study we have found out that when using data in short term to train and test, the RL model performs better. This tells that reinforcement learning should be implemented with precaution for short term duration and after comparing with other artificial intelligent methods models.

While doing this research we have also observed that there is limited study on using artificial intelligence in financial markets so comparing RL with other sophisticated models was not sufficient. Further study on reinforcement learned in future should be done in comparison of RL techniques with other forms of Artificial intelligence methods like deep neural network, recurrent neutral network. In this study we have used historical data to train and test the model. The project has good scope of study in future anyone who wants to analysis further using live trading data, including signals and liquidity of asset.
Acknowledgements

We would like to give a special thanks to Allyn Okun, our SME for this research as well as acknowledge the guidance Dr. Jacquie Cheun-Jensen who led the capstone class this research was accomplished during.

References


**Appendix**

![Figure 11: Middle ranking features of dataset for missing values](image)

https://scholar.smu.edu/datasciencereview/vol7/iss3/6
Figure 12: Bottom ranking features for missing values

Table 1: List of variables included for each ticker symbol in dataset

<table>
<thead>
<tr>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ticker</td>
</tr>
<tr>
<td>SimFinId</td>
</tr>
<tr>
<td>Currency</td>
</tr>
<tr>
<td>Fiscal Year</td>
</tr>
<tr>
<td>Fiscal Period</td>
</tr>
<tr>
<td>Report Date</td>
</tr>
<tr>
<td>Publish Date</td>
</tr>
<tr>
<td>Restated Date</td>
</tr>
<tr>
<td>Shares (Basic)</td>
</tr>
<tr>
<td>Shares (Diluted)</td>
</tr>
<tr>
<td>Revenue</td>
</tr>
<tr>
<td>Cost of Revenue</td>
</tr>
<tr>
<td>Gross Profit</td>
</tr>
<tr>
<td>Operating Expenses</td>
</tr>
<tr>
<td>Selling, General &amp; Administrative</td>
</tr>
<tr>
<td>Research &amp; Development</td>
</tr>
<tr>
<td>Depreciation &amp; Amortization</td>
</tr>
<tr>
<td>Operating Income (Loss)</td>
</tr>
<tr>
<td>Non-Operating Income (Loss)</td>
</tr>
<tr>
<td>Interest Expense, Net</td>
</tr>
<tr>
<td>Pretax Income (Loss), Adj.</td>
</tr>
<tr>
<td>Abnormal Gains (Losses)</td>
</tr>
<tr>
<td>Category</td>
</tr>
<tr>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Pretax Income (Loss)</td>
</tr>
<tr>
<td>Income Tax (Expense) Benefit, Net</td>
</tr>
<tr>
<td>Income (Loss) from Continuing Operations</td>
</tr>
<tr>
<td>Net Extraordinary Gains (Losses)</td>
</tr>
<tr>
<td>Net Income</td>
</tr>
<tr>
<td>Net Income (Common)</td>
</tr>
<tr>
<td>Net Income/Starting Line</td>
</tr>
<tr>
<td>Non-Cash Items</td>
</tr>
<tr>
<td>Change in Working Capital</td>
</tr>
<tr>
<td>Change in Accounts Receivable</td>
</tr>
<tr>
<td>Change in Inventories</td>
</tr>
<tr>
<td>Change in Accounts Payable</td>
</tr>
<tr>
<td>Change in Other</td>
</tr>
<tr>
<td>Net Cash from Operating Activities</td>
</tr>
<tr>
<td>Change in Fixed Assets &amp; Intangibles</td>
</tr>
<tr>
<td>Net Change in Long Term Investment</td>
</tr>
<tr>
<td>Net Cash from Acquisitions &amp; Divestitures</td>
</tr>
<tr>
<td>Net Cash from Investing Activities</td>
</tr>
<tr>
<td>Dividends Paid</td>
</tr>
<tr>
<td>Cash from (Repayment of) Debt</td>
</tr>
<tr>
<td>Cash from (Repurchase of) Equity</td>
</tr>
<tr>
<td>Net Cash from Financing Activities</td>
</tr>
<tr>
<td>Net Change in Cash</td>
</tr>
<tr>
<td>Cash, Cash Equivalents &amp; Short Term Investments</td>
</tr>
<tr>
<td>Accounts &amp; Notes Receivable</td>
</tr>
<tr>
<td>Inventories</td>
</tr>
<tr>
<td>Total Current Assets</td>
</tr>
<tr>
<td>Property, Plant &amp; Equipment, Net</td>
</tr>
<tr>
<td>Long Term Investments &amp; Receivables</td>
</tr>
<tr>
<td>Other Long Term Assets</td>
</tr>
<tr>
<td>Total Noncurrent Assets</td>
</tr>
<tr>
<td>Total Assets</td>
</tr>
<tr>
<td>Payables &amp; Accruals</td>
</tr>
<tr>
<td>Short Term Debt</td>
</tr>
<tr>
<td>Total Current Liabilities</td>
</tr>
<tr>
<td>Long Term Debt</td>
</tr>
<tr>
<td>Total Noncurrent Liabilities</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>Share Capital &amp; Additional Paid-In Capital</td>
</tr>
<tr>
<td>Retained Earnings</td>
</tr>
<tr>
<td>Total Liabilities &amp; Equity</td>
</tr>
<tr>
<td>IndustryId</td>
</tr>
</tbody>
</table>