Petter Kowalik Gran
August Jacob Kjellevold Holm
Stian Gropen Søgård

A Deep Reinforcement Learning Approach to Stock Trading

June 2019

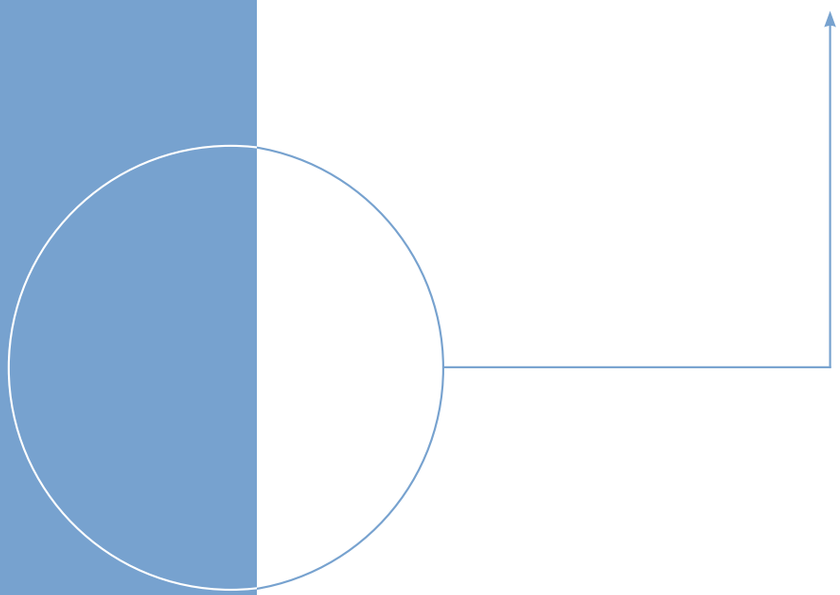
NTNU
Norwegian University of
Science and Technology

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Economics and Management
Department of Industrial Economics and Technology
Management

Master's thesis 2019

Petter Kowalik Gran, August Jacob Kjellevold Holm, Stian Gropen Søgård



NTNU
Norwegian University of
Science and Technology

NTNU

Norwegian University of
Science and Technology

A Deep Reinforcement Learning Approach to Stock Trading

Petter Kowalik Gran
August Jacob Kjellevold Holm
Stian Gropen Søgård

Industrial Economics and Technology Management
Submission date: June 2019
Supervisor: Peter Molnár

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Preface

We would like to express our sincere gratitude to our supervisor, Associate Professor Peter Molnár, for valuable discussions and constructive feedback. Your knowledge, eagerness and constant availability have been essential contributions to the completion of this thesis. Working with you has been a pleasure.

Trondheim, June 11, 2019

Petter Kowalik Gran, August Kjellevold Holm, Stian Gropen Søgård

Abstract

This study investigates the viability and potential of using state of the art Deep Reinforcement Learning for stock trading. We specifically use a *Deep Deterministic Policy Gradient* (DDPG). The model trades stocks in four indices: DJIA (USA), TSX (Canada), JSE (South Africa) and SENSEX (India). We find that DDPG agents using past log return (R) and trading volume (TV) as predictors yield the best performance. The models outperform a buy-and-hold benchmark for all markets in terms of mean return. Adding Google search volume (G) as a predictor does not improve performance in developed markets (USA and Canada), but is valuable in emerging markets (South Africa and India). The algorithm is tested also after implementing transaction cost, where agents are restricted to only trade once every month or quarter. Several agents outperform the benchmark in terms of mean return. Results are compared to a simple linear regression. In terms of mean return, the DDPG agent always outperforms the equivalent linear regressions.

Keywords: Google trends, Deep Reinforcement Learning, Deep Deterministic Policy Gradients, Stock trading.

Sammendrag

Dette studiet undersøker hvorvidt Dyp Forsterkende Læring (eng: Deep Reinforcement Learning) kan brukes til å kjøpe og selge aksjer. Vi implementerer en Dyp Deterministisk Policy Gradient (DDPG)-algoritme (eng: Deep Deterministic Policy Gradient). Algoritmen handler aksjer fra fire forskjellige indekser: DJIA (USA), TSX (Canada), JSE (Sør-Afrika) og SENSEX (India). Resultatene viser at DDPG-agenter som estimerer fremtidig avkasting basert på historisk logaritmisk avkastning (R) og handelsvolum (TV) oppnår best resultater. Disse agentene oppnår en høyere gjennomsnittlig avkastning enn en kjøp-og-hold-portefølje. Det å legge til Google søkevolum (G) som en forklaringsvariabel øker ikke modellens ytelse i velutviklede markeder (USA og Canada), men tilfører verdi i fremvoksende markeder (Sør-Afrika og India). Vi tester også algoritmen med transaksjonskostnader, der agentene er begrenset til å kun handle én gang i måneden eller én gang i kvartalet. Flere av disse agentene får høyere gjennomsnittlig avkastning enn sine referanseporteføljer. Algoritmen sammenlignes videre med en lineær regresjon. Resultatene fra sammenligningen viser at samtlige DDPG-agenter oppnår høyere gjennomsnittlig avkastning enn sin tilsvarende regresjon.

Contents

List of Figures

List of Tables

Abbreviations

AI	=	Artificial Intelligence
API	=	Application Programming Interface
ATV	=	Adjusted Trading Volume
B&H	=	Buy-and-Hold Portfolio
CVaR	=	Conditional Value-at-Risk
DDPG	=	Deep Deterministic Policy Gradient
DJIA	=	Dow Jones Industrial Average
ES	=	Expected Shortfall
FF	=	Fama-French Regression
G	=	Google Trends Data
GA	=	Genetic Algorithm
JSE	=	JSE Limited Top 40
R	=	Log Returns
RL	=	Reinforcement Learning
SENSEX	=	S&P BSE Sensex
SD	=	Standard Deviation
SL	=	Supervised Learning
SR	=	Sharpe Ratio
SVI	=	Search Volume Index
TD	=	Temporal Difference Error
TSX	=	S&P/Toronto Stock Exchange Index
TV	=	Trading Volume
VaR	=	Value-at-Risk

1 Introduction

Prediction of stock returns is possibly one of the most researched topics in finance. The methods grow in sophistication, but the goal remains the same. Early research was mainly concerned with the efficient market hypothesis. Recent advancements in computer hardware have made more computationally intensive analysis possible. With the rise of neural networks, researchers are no longer confined to looking for linear relationships. Deep neural networks applied to the financial markets have gone from being a curious idea to find itself at the very forefront of financial research over the past decade. We want to contribute to the advancement of financial AI by examining the viability of applying deep reinforcement learning to stock markets. Combining reinforcement learning with deep neural networks is a state-of-the-art concept, even in computer science research.

The current hype of deep learning in many ways begun in 2012 when Krizhevsky et al. (2012) crushed the previous top error rate for image classification in the Large Scale Visual Recognition Challenge. The paper introduced principles that have become the core of deep learning. For instance, Facebook uses the same principles to suggest picture tags, and self-driving cars utilize them for object detection.

Krizhevsky et al. (2012)'s extraordinary results exploded the research in deep neural networks. Researchers were attempting to apply the same techniques practically everywhere. Mauro (2016) connected to the Science Direct API and discovered that the growth in the number of research papers concerning deep learning went from virtually zero to the highest growth among important AI topics[1] in 2013. Common for these topics is that they focus on supervised learning, but other approaches exist, such as reinforcement learning.

In 2015 Mnih et al. (2015) published a paper where a combination of reinforcement learning and deep neural networks was used to build an agent that was tested on a challenging domain of classic Atari 2600 games. The deep Q-network agent they built only received the pixels and game scores as input and achieved a performance level comparable to that of a professional human gamer, surpassing the performance of all previous algorithms. Building on this research, Silver et al. (2016) developed an agent using the same principles. This model defeated a professional Go[2] player 5 out of 5 games, a feat thought to be a decade away. A year later, Silver et al. (2017) built yet another agent, this time using absolutely no human knowledge during its training. The upgraded agent beat the previously published agent (Silver et al., 2016) in 100 out of 100 games. The agent's ability to perform on an extremely complex domain, without any human knowledge

[1] The included topics: Deep Learning, Support Vector Machine, Neural Networks, Data Mining, Speech Recognition, Image Recognition, Recommender System

[2] The game of Go has long been viewed as the most challenging of classical games for artificial intelligence due to its enormous search space and the difficulty of evaluating board positions and moves.

as input, led to increased interest in testing the approach in other domains.

Using supervised learning with deep neural networks for prediction in financial markets has been extensively tested, and the technique has produced very good results. Heaton et al. (2016), among others, show that deep neural networks have the potential to, sometimes drastically, improve predictive performance in conventional financial applications. Unfortunately, even in successful applications of machine learning in finance, certain issues and concerns exist. First of all, deep learning methods generally have low interpretability. Additionally, the widespread use of opaque models can result in unintended, possibly negative, consequences when models' action interact in the marketplace (Financial Stability Board, 2017). Furthermore, Fischer and Krauss (2018) find that the predictive power of neural networks disappears when applied to financial data from the year 2010 and onwards. They argue that an increase in usage of deep supervised learning on financial markets has removed any potential arbitrage. However, with deep reinforcement learning showing high potential in solving problems where supervised learning fails, it is possible that a similar approach to that of Silver et al. (2017) in the stock market could create a profitable trading strategy.

Google's DeepMind launched the formidable AlphaZero agent in October 2017 (Silver et al., 2017). So far there has not been much research on applying the same models in finance. Deng et al. (2017) and Huang (2018) utilize similar methods, but they use a discrete search space. Stock trading, however, has a continuous search space since positions can take continuous values. Even though there has been extensive research on using machine learning in finance, see chapter 2, not much has been done combining deep learning and reinforcement learning. We have identified three unpublished papers that use a continuous search space, Jiang et al. (2017), Filos (2018) and Liang et al. (2018). However, these papers are not peer-reviewed publications. Thus, using this methodology on financial data is to a large extent unexplored territory.

Inspired by Deepmind's AlphaZero we create a deep reinforcement learning agent for stock trading and test it on four stock indices in four countries. We find that it is able to outperform a buy-and-hold benchmark in all markets, also after imposing transaction cost. Furthermore, a positive alpha, significantly different from that of the benchmark portfolio, is achieved in three out of four markets. The findings suggest that deep reinforcement learning could be a viable technique for financial applications.

2 Literature Review

2.1 Deep Learning in Finance

Deep learning is becoming increasingly popular in financial studies. Still the potential seems enormous. Chourmouziadis and Chatzoglou (2016) predict that deep learning will play a key role in future financial time series forecasting. Predicting stock prices or returns using neural networks is the main application of interest. According to Chong et al. (2017) there are two main approaches for making predictions with neural networks: either straight forward based on traditional financial data, or by analyzing financial news feeds using recurrent neural networks.

The first approach is well studied. Papers like Freitas et al. (2009), Niaki and Hoseinzade (2013), Arévalo et al. (2016), and Heaton et al. (2016) all use historical financial data to predict future price movements. Most of these studies report positive results, beating relevant benchmarks. The second approach is less studied, but relevant publications are Ding et al. (2015) and Oshihara et al. (2014). Also here the results are promising.

The aforementioned research is based on supervised learning. Training examples are labelled as either good or bad based on whether the asset value increased or decreased. However, a different approach exists. In reinforcement learning, labeled examples are not needed, and the methodology can be viewed as more general. Effectively, an implementation is not restricted to a specific market or financial instrument.

Some research on using reinforcement learning in financial applications has been conducted. However, the studies largely look at a discrete action space, and thus output a discrete trading signal per asset. Examples include Moody and Saffell (2001), Dempster and Leemans (2006), and Cumming (2015), along with Deng et al. (2017) and Huang (2018) in more recent years.

Stock trading, though, is a continuous problem, and therefore the aforementioned models are not well suited. Deep reinforcement learning, using DDPG, offers a model free, machine-learning based approach to the challenge of trading stocks, allowing for continuous state and action spaces. The research on this field, for financial applications, is largely non-existent most likely due to the quite recent advancements in the field.

There are some, unpublished, recent papers that implement a similar method. Most notably Jiang et al. (2017) use the DDPG framework to trade a number of cryptocurrencies and appear to get good results. Filos (2018) tries the same trading the S&P500 components. Liang et al. (2018) also test the framework, but are more sober in their analysis of the algorithm's performance. Benchmarking against previous studies is complicated in itself for this method. Performance is

highly dependent on the choice of hyperparameters, and the stochasticity of the environment according to Islam et al. (2017). Therefore, results can be quite difficult to reproduce.

2.2 Genetic Algorithms in Finance

Genetic algorithms have been utilized in a variety of ways in financial studies. Applications range from macroeconomic simulations (Arifovic, 1994, 2001) through trading rules development (Allen and Karjalainen, 1999; Neely et al., 1997; Neely and Weller, 1999; Wang, 2000; Dempster and Jones, 2001) to portfolio optimization (Jevne et al., 2012; Chang et al., 2009; Lin and Liu, 2008; Tsao and Liu, 2006; Krink and Paterlini, 2011; Soleimani et al., 2009).

Exploiting trading rules created by genetic algorithms have yielded varied results. Neely et al. (1997), who look at foreign exchange markets, find strong evidence of economically significant out-of-sample excess returns for all considered currencies. Neely and Weller (1999) are also able to achieve excess returns, albeit only for three out of four considered currencies in the 1986-1996 testing period. Dempster and Jones (2001) are more cautious in their conclusion, but still claim that results indicate a significantly profitable strategy in the presence of realistic transaction cost.

Allen and Karjalainen (1999) use a genetic algorithm to find trading rules for the S&P500 index. Contrary to the evidence from FX markets, the strategy does not give excess returns after transaction cost. Wang (2000), looking at the S&P500 spot and future markets, support the findings of Allen and Karjalainen (1999). They are unable to find trading rules that consistently outperform the market post transaction cost.

Genetic algorithms in financial research is perhaps most widely used in the realm of portfolio optimization. Conventional methods, such as quadratic programming, often rely on unrealistic assumptions like linearity. Such simplifications are not necessary when using evolutionary methods (Krink and Paterlini, 2011). Jevne et al. (2012); Lin and Liu (2008); Tsao and Liu (2006); Tsao (2010) all conclude that standard mean-variance optimization methods are inefficient, and a better solution can be obtained applying genetic algorithms to the respective problems.

Evolutionary techniques have also been well tested in other domains. Mahfoud and Mani (1996) generate excess returns based on single stock forecasting, looking at 1600 individual stocks. Shin and Lee (2002) achieve promising results in bankruptcy prediction, and Kim and Han (2000) train an artificial neural network using genetic algorithms that outperforms its regularly trained counterpart by 11% on the Korean stock market.

Applying genetic algorithms to financial applications have shown promise, and the versatility of the algorithm framework inspired us to make it part of our method.

2.3 Google Trends in Finance

Google search data is a widely used proxy for information demand. Several papers have investigated the predictive power of Google search data based on this fact. Similar measures of information demand are analyst coverage and firms' changes in advertisement expenses, as suggested by Chen et al. (2018). However, Google search volume is the preferred measure for two main reasons. First of all, Google is by far the most used search engine in the world. According to StatCounter Global Stats, as of May 2019, 92.0% of internet search queries are made through Google. Secondly, and perhaps more importantly, a Google search undoubtedly reveals interest in the queried topic (Harford, 2017).

Search volume data is an extremely versatile data type. Consequently, Google search data (Google trends) has been used to predict everything from disease outbreaks (Eysenbach, 2006; Polgreen et al., 2008; Ginsberg et al., 2009; Carneiro and Mylonakis, 2009; Pelat et al., 2009), through travel destination planning, consumer confidence, automobile sales (Choi and Varian, 2012), to gasoline prices (Molnár and Bašta, 2017).

Attention, as a scarce cognitive resource, can affect asset prices (Kahneman, 2013). Furthermore, Da et al. (2011) and Joseph et al. (2011) suggest that the search volume index of Google (SVI) can summarize investor attention and sentiment in a significant way. This, along with increasing interest in Google trends from the scientific community, provides motivation for using Google trends in financial research.

In the early stages of financial Google trends research, most attention was given to the U.S. markets. More recently, studies of other markets have become abundant. For instance, Aouadi et al. (2013) investigate volatility in the French market, Takeda and Wakao (2014) study financial instruments in Japan, Kim et al. (2018) conduct research on the Oslo stock exchange, while both Bank et al. (2011) and Fink and Johann (2013) look at trading activity in the German stock market.

Financial studies using Google trends search data have mainly focused on predicting stock prices and returns, trading volume, and volatility. Some papers consider only an aggregate market index, such as Challet and Ayed (2013) and Tantaopas et al. (2016), while the majority of papers study markets with multiple assets. For studies on multiple assets, search volume seem to be useful for trading volume predictions (Preis et al., 2010; Bank et al., 2011; Vlastakis and Markellos, 2012; Fink and Johann, 2013; Aouadi et al., 2013; Takeda and Wakao, 2014; Kim et al., 2018; Aalborg et al., 2018).

Preis et al. (2013) indicate a relation between search volume for keywords related to finance, and corresponding market volatility. Furthermore, Vlastakis and Markellos (2012); Aouadi et al. (2013); Da et al. (2014); Goddard et al. (2015); Dimpfl and Jank (2016); Kim et al. (2018) all suggest that investor sentiment can predict future asset volatility. Interestingly, very few papers

conclude the opposite.

The scientific community is divided in the question of whether Google trends search data can be used to predict stock prices and returns. Da et al. (2011); Joseph et al. (2011); Preis et al. (2013); Da et al. (2014); Vozlyublennaia (2014); Gwilym et al. (2016) claim to find a relation between search volume and future returns, either by identifying immediate returns or by finding mean reversal patterns. Preis et al. (2010); Takeda and Wakao (2014); Kim et al. (2018), on the other hand, were unable to identify a significant relationship between investor attention and returns or prices. Some studies develop trading strategies based on predicting stock returns with search volume. The strategies seem to beat relevant benchmarks such as market indices or buy-and-hold strategies (Preis et al., 2013; Kristoufek, 2013; Bijl et al., 2016; Hu et al., 2018). These results motivate us to investigate whether search volume data can be exploited by a sophisticated deep reinforcement learning algorithm to produce a profitable trading strategy.

3 Data

Our model uses daily stock log returns, daily stock trading volume and daily Google search data. Trading volumes have historically been used as an indicator of investor sentiment (Karpoff, 1987; Campbell et al., 1992; Wang et al., 2018). As explained in chapter 2, more recently Google search data have complemented trading volumes as it represents a very direct measure of attention. The sample period is from January 1. 2004 until December 31. 2018. The initial training period starts in 2004, testing starts in 2008. We test the model on every quarter between 2008 and 2018. The model is updated before each quarter according to the training/testing scheme described in chapter 4. Since Google search volume is to be transformed relative to its past, and Google only offers data from 2004 and onwards, the initial training period becomes 2005-2008.

3.1 Selected Markets

We consider four stock markets, differing in both geographic location and descriptive statistics. Common for them all is the use of English as the official language in the country, a condition in order to properly use Google trends data in the analysis. Also, Google's search engine market share is above 90% in every country[1].

We study the Dow Jones Industrial Average (USA), the SENSEX (India), the TSX60 (Canada), and the JSE Limited Top 40 (South Africa). To have comparable results across markets, the number of stocks from each index must be similar. Companies not listed for the whole training and testing period are filtered out. Companies with insufficient Google trends data are removed. We are left with approximately 30 stocks from each market.

3.2 Survivorship Bias

An index' component stocks usually change over time. A stock can be dropped because it does not meet the index' requirements, such as rule compliance, because it no longer attracts investor attention, or simply if the stock is taken private. Consequently, some company turnover is to be expected. This is the case for all stock indices considered in this thesis. The company turnover creates a survivorship bias in the remaining stocks; the stocks that have survived for the whole period can be considered "winners", and are likely to have somewhat different statistical properties from the ones that were removed.

Two main problems arise from this consideration. Firstly, benchmarking against the index itself

[1] StatCounter, April 2019

becomes less meaningful, since the index stock components change while the sample of stocks we analyze does not. Secondly, as we are looking at a sample with survivorship bias, abnormal returns cannot be fully confirmed simply by finding a significant positive alpha in a Fama-French regression.

We tackle the first challenge by not benchmarking against the indices directly, but rather against buy-and-hold portfolios consisting of the same stocks our algorithm uses for training and testing. An alternative solution to using a B&H benchmark could be to continuously update the considered stocks during training/testing to track the index. The solution is not applicable to our method because the algorithm learns how to trade specific stocks, see chapter 4, and not stock trading in general. The second challenge is dealt with by checking whether a positive alpha is significantly different from the B&H portfolio alpha.

3.3 Transformation of Stock Data

Adjusted daily closing prices, along with daily trading volumes, are obtained from Yahoo! Finance. For any given date, if any of the component stocks are missing a value, the whole date row is excluded for all the stocks. Price data and trading volume data are transformed in order to improve model performance.

We transform the price data relative to its past simply by taking

$$R_t = log\Big(\frac{P_t}{P_{t-1}}\Big) \tag{3.1}$$

where R_t is the log return, P_t is the adjusted stock close price for day t and P_{t-1} is the adjusted stock close price for day $t-1$. We use log returns throughout the study.

At time t, the log returns for a stock are normalized relative to other other stocks by taking

$$R^*_{i,t} = \frac{R_{i,t} - \overline{R}_{0:n,t}}{SD(R_{0:n,t})} \tag{3.2}$$

where $R_{i,t}$ denotes the log return of stock i at time t, $\overline{R}_{0:n,t}$ denotes the average log return of all stocks at time t, and $SD(R_{0:n,t})$ denotes the standard deviation of the log return of all stocks at time t. This normalization process is supported by our own empirical results from initial testing.

The relative value of daily trading volume is dependent on the year the stocks were traded. Therefore, some standardization relative to the past must be performed. We transform the daily trading volume using

$$ATV_t = \frac{TV_t - \frac{1}{252}\sum_{i=1}^{252} TV_{t-i}}{SD(TV, t)} \tag{3.3}$$

where ATV_t is the adjusted daily trading volume, TV_t is the daily trading volume at time t, and

$SD(TV, t)$ is the standard deviation of the TV_t for the past 252 days. The window of 252 days is chosen to reflect the number of business days in a year. This is a standard transformation, for instance used by Kim et al. (2018). After adjusting trading volume relatively to the past, we perform the same normalization procedure as with returns (3.2). At time t, the trading volume becomes

$$TV_{i,t} = \frac{TV_{i,t} - \overline{TV}_{0:n,t}}{SD(TV_{0:n,t})} \tag{3.4}$$

where $TV_{i,t}$ denotes the trading volume of stock i at time t, $\overline{TV}_{0:n,t}$ denotes the average trading volume of all stocks at time t, and $SD(TV_{0:n,t})$ denotes the standard deviation of the trading volume of all stocks at time t.

3.4 Google Trends

Google trends is a real-time daily index of the volume of queries users enter into the Google search engine (Kim et al., 2018). The platform also gives access to what it calls *non-real time data*, which pertains to historical data from 2004 up to 36 hours in the past. Google trends uses a standardized scale of 0 to 100 where 100 represents the highest search volume (SVI) during a considered time period and geographic region (Choi and Varian, 2012). Stock returns and trading volume is only available for Monday through Friday each week, while Google trends data is available for all seven days. Therefore, we average searches from Saturday, Sunday, and Monday when calculating total search volume for Monday by taking

$$S_t = \frac{x_t + x_{t-1} + x_{t-2}}{3} \tag{3.5}$$

where x_t is the raw search volume for a given stock on Monday and S_t is the average search volume for Monday and the weekend.

From Google, daily data for a given stock can be downloaded in bulks of maximum 270 days. We downloaded the Google trends data through a publicly available pseudo API[2]. Subsequently, we concatenate the 270-day-bulks to create a complete time series of daily search volume for the period 2004-2018. Note that Google trends data is computed using a sampling method. Consequently, the results from an identical query ran several times most likely varies slightly.

The value of the raw Search Volume Index (SVI) is dependent on the time period the data was downloaded. Some standardization relative to the past must therefore be performed. After the concatenation into a complete time series, we transform the SVI as we did with trading volumes by taking

$$A_t = \frac{S_t - \frac{1}{252}\sum_{i=1}^{252} S_{t-i}}{SD(S, t)} \tag{3.6}$$

where A_t is the abnormal SVI, S_t is the SVI at time t after accounting for weekends, and $SD(S, t)$

[2]https://github.com/GeneralMills/pytrends

is the standard deviation of the S_t for the past 252 days. As when transforming trading volumes, the constant window of 252 days is chosen to reflect the number of business days in a year. Since Google trends data are only available from 2004 we cannot transform the first 252 days with our method (equation 3.6). Thus, the training period starts in 2005.

As with log returns (3.2) and trading volume (3.4), for a stock, we normalize the Google trends data relative to other stocks. The normalized values become

$$G_{i,t} = \frac{A_{i,t} - \overline{A}_{0:n,t}}{SD(A_{0:n,t})} \tag{3.7}$$

where $A_{i,t}$ is the abnormal search volume after transformation for stock i at time t, $\overline{A}_{0:n,t}$ denotes the average abnormal search volume for all stocks at time t, and $SD(A_{0:n,t})$ is the standard deviation of the abnormal search volume of all stocks at time t.

We employ a filtering process to decide what search words to use when collecting Google trends data. Company names are preferred over tickers since Bijl et al. (2016) conclude that company name searches have stronger relationships with stock market returns. Common words such as 'ltd.', 'inc.', 'company' are excluded from the company name. Tickers are used if the company name is too general or if a company name does not give results due to insufficient search volume. Google trends allow for geographical filtering. Preis et al. (2013) found that data filtered according to a geographic location can better explain movements for that specific geographic location. Thus, this study only considers Google searches from the country in which the stock exchange is based. Google also allows of filtering on categories, of which Finance is one. Bijl et al. (2016) conclude that the Finance filter does not provide any improvement over unfiltered searches in terms of predicting stock returns. Thus, we do not use this filtering option. See appendix A.2 for the full list of search words.

4 | Methodology

In this study, we define three trading strategy models: reinforcement learning, linear regression and buy-and-hold. The strategies based on linear regression and buy-and-hold are used for benchmarking purposes. Additionally, the Fama-French model is used to check whether a positive alpha is significantly larger than that of the benchmark.

4.1 Reinforcement Learning

Our agent applies reinforcement learning to stock trading. In traditional reinforcement learning there is no blueprint. Right or wrong choices are not clearly labeled like in a supervised learning framework. Ultimately, learning becomes a process of trial and error. This situation has similarities to how humans learn. We illustrate the reinforcement learning process with the real-life example of learning to ride a bike. It is not something one can learn by reading a book or being told how to do in theory. Rather it is the results of your actions that make you learn. If you fall off the bike, you know you did something wrong. Similarly, you know you are doing things right if you are able to ride increasing distances without falling off.

The analogy of learning to ride a bike can illustrate how our agent learns to trade stocks through trial and error. At each time step, the agent proposes a set of portfolio weights. As in the biking example, the agent subsequently learns by experiencing the result of its actions. A decreasing portfolio value is equivalent to falling off the bike, an increase corresponds to not falling off. More formally, the reward function is the log return of the portfolio at the end of each episode such that it maximizes

$$\max z = \sum_{t=1}^{t=T} log(\frac{PV_t}{PV_{t-1}}) \tag{4.1}$$

where T is the number of trading days within an episode, and PV_t is the portfolio value at time t with $PV_0 = 1.0$. We test agents with four different episode lengths: one day, one week, one month and one quarter. Designing an appropriate reward function is a challenge in reinforcement learning according to Sutton and Barto (2018), therefore learning risk-adjusted behavior is not as simple as defining the reward function to be portfolio Sharpe ratio or another risk-adjusted metric. Without any clear alternatives, simple log returns can still be an interesting reward function to test.

Most people have experience with learning to ride a bike and know how hard it can be. The same holds for traditional reinforcement learning; it can be hard to learn good strategies merely through a simple trial and error method. As in learning to ride a bike, it helps to have somebody that can give you feedback on your actions, like a parent. He or she might tell you where to keep

your eyes, or how to distribute your weight. Our agent employs the same scheme by relying on two major components: an actor and a critic. The actor outputs an action (a set of suggested portfolio weights) at each time step. In the bike analogy, the actor would be the person learning to ride a bike. The critic, being the parent, takes the action of the actor as input and predicts the quality of that action to the best of its knowledge. The actor takes that prediction and uses it as feedback for learning. Subsequently, the critic sees the actual result of the action, whether the portfolio value is increased or decreased, and uses it to improve its prediction skills. So instead of learning to ride a bike by falling off it, you learn using the feedback from your parent. The parent's feedback accuracy is then improved by watching whether you fall off the bike or not.

Sutton and Barto (2018) define *deep* reinforcement learning as reinforcement learning where both the actor and critic are modeled by neural networks. Even though we utilize neural networks, our reinforcement learning method is not to be confused with less complex supervised learning methods, such as Niaki and Hoseinzade (2013), Freitas et al. (2009), and Heaton et al. (2016). Still, there are certain similarities. First of all, our method is a stochastic local search algorithm, which means that during training, up until convergence, the agent can yield different results on the same data set. There is a trade-off between optimality and generality. Usually, the chosen training the length of the training phase is longer than the average time taken for the model to converge. The time to convergence can be defined as the number of epochs, an epoch is defined as one full iteration over all available training episodes, needed for the agent to choose the same action on the same input data across epochs. Training is time-consuming. Naturally, it is essential to minimize the number of epochs while ensuring convergence. For our datasets, 300 epochs seems to be an appropriate number of epochs for the model to converge. Secondly, the methodology is a black-box. Consequently, analyzing how and why the algorithm makes its decisions is highly challenging. For example, when Silver et al. (2016) defeated the European champion in Go, the algorithm made several moves that no human Go expert could make sense of during the game, but were deemed decisive for the final outcome in retrospect.

The next section provides further detail about the methodology, focusing on mathematics and specific implementation details.

4.2 Deep Deterministic Policy Gradients

Our agent is based on the DDPG algorithm (Silver et al., 2016). DDPG is a policy gradient method that uses a stochastic behavior policy for exploration but estimates a deterministic target policy. The algorithm has two main steps. First, the current policy is evaluated. Then, the performance is maximized by following the policy gradient. The algorithm uses an actor-critic scheme. Two neural networks are used, one for the actor and one for the critic. The actor estimates the policy function $\mu_\theta(s)$ by taking the state, s, as input and returning an action chosen from the continuous action space. The critic takes that action and the state and calculates a

temporal difference error, $Q(s, a|\theta^Q)$. The output of the critic is used to train both the actor and the critic network (Lillicrap et al., 2015). Figure 4.1 illustrates the procedure.

The update rule for the actor parameters is provided by the deterministic policy gradient theorem (Silver et al., 2014): suppose the portfolio management problem satisfies the appropriate conditions found in Silver et al. (2014). The gradients of the actor and the critic exist, which implies that the deterministic policy gradient exists. Let $\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)}$ be the gradient of the critic with respect to the actions and let $\nabla_{\theta^\mu}\mu(s|\theta^\mu)_{s=s_t}$ be the gradient of the actor with respect to its parameters. The deterministic policy gradient $\nabla_{\theta^\mu}\mu$ is defined as (Emami, 2016):

$$\begin{aligned}\nabla_{\theta^\mu}\mu &\approx \int_S \rho^{\mu'}(s_t)\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)_{s=s_t}ds \\ &= E_{\mu'}[\nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)_{s=s_t}]\end{aligned} \tag{4.2}$$

By estimating a deterministic target policy, we can update the parameters of the actor by moving them in the direction of the action-value function gradient (Lillicrap et al., 2015):

$$\theta^\mu_{k+1} = \theta^\mu_k + \alpha E_{\mu'^k}[\nabla_\theta Q(s, \mu(s|\theta^\mu_k)|\theta^Q_k)] \tag{4.3}$$

We apply the chain rule to get

$$\theta^\mu_{k+1} = \theta^\mu_k + \alpha E_{\mu'^k}[\nabla_a Q(s, a|\theta^Q_k)|_{a=\mu(s|\theta^\mu_k)}\nabla_\theta \mu(s|\theta^\mu_k)] \tag{4.4}$$

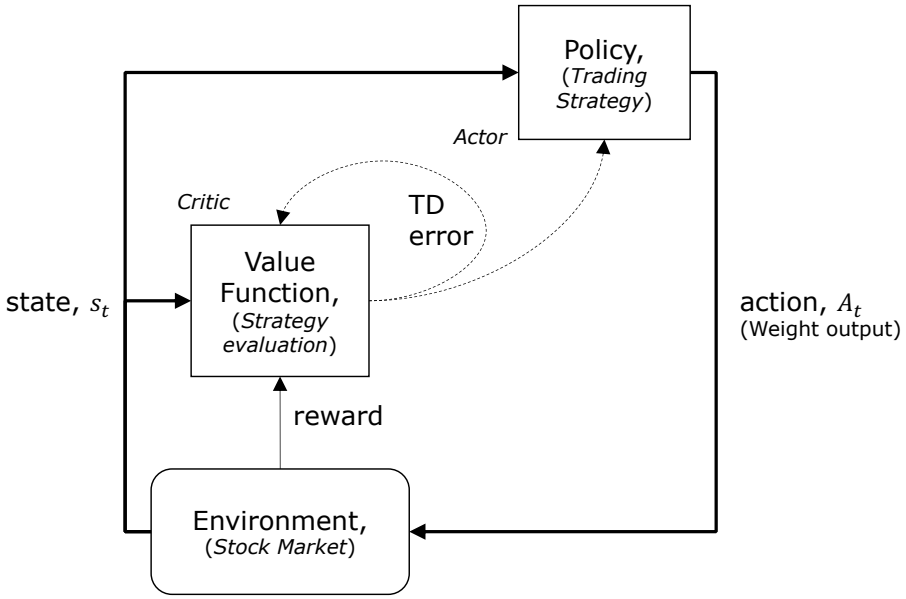


Figure 4.1: Overview of the DDPG architecture.

Applying non-linear function approximators such as neural networks to deterministic policy gradients can make the algorithm unstable (Mnih et al., 2015). Based on Lillicrap et al. (2015),

we stabilize the DDPG algorithm using two strategies: the use of *experience replay* and dedicated *target networks*.

During training, the algorithm stores each observation along with the chosen action, new state, and a received reward, in a replay buffer. The probability of getting stuck on a local optimum is reduced by re-training on randomly pulled samples of observations from the replay buffer. This procedure is known as experience replay (Mnih et al., 2015).

Dedicated target networks are networks that are no longer subject to training. Following a fixed scheme, the weights of the actor and critic networks are copied into new networks. For the next interval of training, these networks provide target values for the actor and critic to estimate. Target networks preserve the distribution of data and provide more consistent targets during training (Mnih et al., 2015).

The reinforcement learning problem is fully described by the following components: the state space S_t, the action space A_t and the reward function $R_a(s, s')$. We define S_t as an $(m \times n)$ vector, where m is the number of assets, and n is the number of asset features considered. A_t is an $m \times 1$ vector of portfolio weights w_{it}, where w_{it} is a real number between zero and one. Additionally, we require that $\sum_i w_{it} = 1$, for all t, and the immediate reward at time t is the log of the portfolio value, V_t, at time t divided by the portfolio value in the previous time step, V_{t-1}.

The DDPG algorithm implementation closely resembles that of Lillicrap et al. (2015), but with a few extensions. Specifically, we use an online learning scheme and a portfolio weight memory. See table A.1 and algorithm 1 for implementation details.

Online learning means that the agent continuously improves itself, also during the testing phase. Immediate data can be more important than older observations when attempting to predict the future (Zhang, 2003). Online learning helps the agent adapt to more recent trends.

We also extend the model by using a portfolio weight memory. All the weights in the initial portfolio are set to zero, except cash which is set to one. The weights of the current portfolio are fed into the final layer of the actor network. Portfolio weight memory helps the actor network remember its current position in the market.

Algorithm 1: Pseudo code for DDPG Trading Algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s, a|\theta^\mu)$ with weights θ^Q and θ^μ;
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$;
Initialize replay buffer R;
Fill replay buffer with samples from genetic algorithm (see section 4.3);
for *episode = 1,M* **do**
    Receive initial observation state s_0;
    for $t = 1, T$ **do**
        Receive current portfolio a_{t-1};
        Select action $a_t = \mu(s, a_{t-1}|\theta^\mu)$ according to the current policy;
        Execute action a_t and observe reward r_t and observe new state s_{t+1};
        Store transition $(s_t, a_t, a_{t-1}, r_t, s_{t+1})$ in R;
        Sample a random minibatch of N transitions $(s_i, a_i, a_{i-1}, r_i, s_{i+1})$ from R;
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$;
        Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$;
        Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)_{s=s_t}$$

        Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

    end
end

4.3 Pre-training with Genetic Algorithm

An issue when implementing reinforcement learning algorithms is to handle the explore-exploit dilemma. The agent learns trading rules by exploring the action space; the challenge is to decide when to shift the computational resources from the learning process to solving the actual problem (Sutton and Barto, 2018). To enhance the learning process, we use a pre-training scheme where a genetic algorithm fills the replay buffer with a variety of examples that represent different trading behavior.

Genetic algorithms are inspired by evolution. A potential solution to a problem is called an individual. A population is a set of individuals. An individual has a genetic base, which

represents the main features of that specific potential solution. A fitness function calculates how well the genetic base solves the problem.

During pre-training, we initialize a population of individuals. Subsequently, a new population is created through reproduction among the initial individuals. During reproduction, a new individual is created by combining the genetic material, the preliminary solutions, of two parent individuals. The parents are more likely to be chosen for reproduction if they have a high fitness value. This process repeats for a given number of generations, or until a target fitness is reached.

In our specific implementation, an individual's genetic base represents a trading strategy. For each quarter in the DDPG training data set, the genetic base decides what portfolio to hold. Initially, the individuals' respective positions are random. During reproduction, a "cut-off" quarter is randomly selected. The child individual copies the trading strategies of parent 1 up until this quarter, and the strategies of parent 2 after this quarter. Genetic mutation happens by picking an arbitrary quarter and randomly altering the child's position for that specific quarter.

Traditionally when genetic algorithms are applied to portfolio optimization the fitness function corresponds to objective functions such as Sharpe ratio, VaR or CVaR. However, the goal of our genetic algorithm is not to maximize return or any risk-adjusted measure. Instead, we want to help the model obtain a good balance between exploration and exploitation. Therefore, we choose a non-traditional, but more suitable, fitness function.

$$Fitness = R - Variance + Variety \tag{4.5}$$

where R is the return the individual's strategy obtains, $Variance$ is the variance of the returns of said strategy, and $Variety$ is a measure for the uniqueness of the given strategy compared to the rest of the population. Variety is comprised of two components: the number of positions the individual holds and the individual's preferred stocks. The larger the differences compared to the rest of the population, the higher the variety score. The variety term in the fitness function ensures that the pre-training scheme sufficiently explores the action space. After the genetic algorithm converges, the trading strategies of the final population fill the initial DDPG replay buffer.

4.4 Training and Testing Scheme

To train and test the model we implement a version of Walk-forward testing (Kirkpatrick II and Dahlquist, 2010). This ensures that we only test our model on out-of-sample data and that the model sees as many training examples as possible. Our out-of-sample period is always one quarter, and the model is trained from 2005 up until the quarter to be tested. Within each calendar year, we simply update the model with information from the latest quarter instead of resetting the model completely and starting from 2005 again. For example, when testing 2012Q3,

the model is fully trained from 2005-2011, and updated with data from 2012Q1 and 2012Q2. This method saves computational time and leads to only very minor loss of performance. This whole process is repeated for all quarters in the 2008-2019 testing period. With this scheme, the algorithm uses about 24 hours for training and testing one agent. Figure 4.2 is a graphic representation of our training/testing scheme.

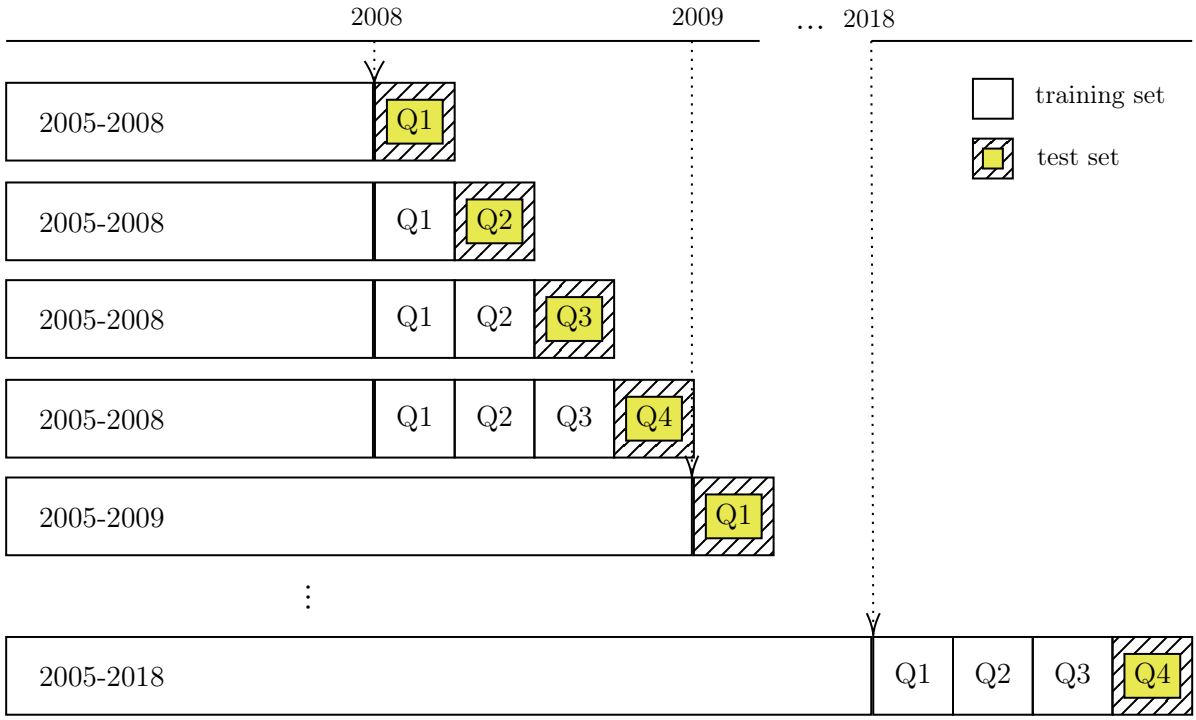


Figure 4.2: Walk forward testing.

4.5 Benchmarks

In addition to the regular buy-and-hold portfolio, we benchmark against a simple linear regression. We calculate the Fama-French Five-Factor intercept for each model and check whether the alpha is significantly larger than that of the buy-and-hold benchmark.

4.5.1 Linear Regression

We utilize a fixed effects panel data model. We let R_{it} be the daily log return, TV_{it} the daily trading volume and G_{it} the daily Google trend search volume for stock i. The linear regression agent estimates the following model

$$\begin{aligned} R_{i,t+1} = \alpha_i &+ \beta_1 R_{it} + \beta_2 \bar{R}_{i,t-5:t} + \beta_3 \bar{R}_{i,t-22:t} + \beta_4 \bar{R}_{i,t-66:t} + \\ &\beta_5 TV_t + \beta_6 \bar{TV}_{i,t-5:t} + \beta_7 \bar{TV}_{i,5-22:t} + \beta_8 \bar{TV}_{i,t-66:t} + \\ &\beta_9 G_{it} + \beta_{10} \bar{G}_{i,t-5:t} + \beta_{11} \bar{G}_{i,t-22:t} + \beta_{12} \bar{G}_{i,t-66:t} + \epsilon_{it} \end{aligned} \tag{4.6}$$

where $\bar{X}_{t-h:t} = \frac{1}{h}\sum_{j=1}^{h} X_{t-j}$ is the the h-day average of variable X_t. The variables reflect the daily, weekly, monthly and quarterly log returns, trading volume and Google search volume.

We conduct out-of-sample trading by first using the regression to predict a one-day-ahead return for each stock, buying the best stock based on said prediction, and repeating the process for each time step. Making a common panel data regression model, and choosing the top 1 stock at each time step, closely resembles the behavior of the reinforcement learning algorithm.

4.5.2 Fama-French

To test whether the DDPG strategy yields abnormal returns we run a Fama-French Five-Factor regression (Fama and French, 2014) using

$$r - r_f = \alpha + \beta(r_m - r_f) + \beta_{smb}SMB + \beta_{hml}HML + \beta_{rmw}RMW + \beta_{cma}CMA + \epsilon \quad (4.7)$$

It is important to note that by using the current DJIA component stocks we have to consider the survivorship bias problem discussed in section 3. We check whether two alphas from portfolios A and B, α^A and α^B, are significantly different from each other by regressing

$$r^A - r^B = \alpha^{AB} + \beta^{AB}(r_m - r_f) + \beta^{AB}_{smb}SMB + \beta^{AB}_{hml}HML + \beta^{AB}_{rmw}RMW + \beta^{AB}_{cma}CMA + \epsilon^{AB} \quad (4.8)$$

where r^A and r^B are the returns of portfolio A and portfolio B. If α^{AB} is significant, we can say that α^A is significantly different from α^B.

5 Results

We assess the viability and potential of the DDPG algorithm applied to finance by letting the agent trade component stocks of major stock indices across the world. The considered markets are United States, Canada, South Africa and India. We run the algorithm on a long testing period, including the financial crisis of 2008, and on several markets. This makes it less likely that results are caused by chance, rather than by the algorithm.

For each market, we compare DDPG performance to that of the equivalent buy-and-hold portfolio. Performance assessment is based on a range of financial metrics. We check for positive abnormal returns using Fama-French analysis (Fama and French, 2014). For the markets where country specific Fama-French factors are unavailable, we use regional factors. By using the current index component stocks we have to carefully consider the survivorship bias problem discussed in chapter 3. A positive alpha is not conclusive evidence for abnormal returns since the tradable stocks are the current survived index components and not a random selection. We deal with this issue by checking whether the alpha is significantly different from the relevant buy-and-hold benchmark, see section 4.5.2.

The model is tested using different sets of predictors: past log returns (R), past trading volume (TV), past Google trends data (G), and all combinations of the three. We investigate what predictors the DDPG algorithm seems most responsive to, and whether or not the DDPG predictor preference is constant across all considered markets.

We implement a model that accounts for transaction cost in order to gauge actual performance more realistically, maintaining the buy-and-hold portfolios as a benchmark. Additionally, we run analysis on less general versions of the DDPG where the agents are restricted to trade only every week, month or quarter. These agents are compared to a linear regression model, in addition to the regular buy-and-hold portfolio. Finally, we present a detailed analysis of algorithm behavior.

5.1 Explanation of Financial Metrics

Mean return is the average log return over the testing period. Volatility (standard deviation), Skewness and Kurtosis are the second, third and fourth moments respectively of a random variable X, calculated on daily log returns.

We look at 5% and 95% Value at Risk (VaR), which are the daily percentage losses not exceeded with probability 5% and 95% respectively. Additionally, we compare values for 5% Conditional Value at Risk ($CVaR$). The metric tells us the average absolute value of the daily returns lying in the bottom 5%. We define 95% $CVaR$ as the equivalent value for the top 5% returns.

Hit ratio is the percentage of the daily log returns that are positive, over the whole testing period. We also use average daily positive returns and the average daily negative returns.

Sharpe Ratio (SR) is used as a measure of risk-adjusted returns. This evaluation metric is mainly for illustrative purposes, as volatility is not included in the algorithm's reward function (equation 4.1). Additionally, the intercepts from Fama-French Five-Factor regressions are used to check for abnormal returns by comparing DDPG agent alpha to B&H alpha and checking whether the difference is significant.

5.2 DDPG Performance on DJIA, TSX, JSE and SENSEX

The algorithm is tested using various sets of predictors: past log returns, past trading volume, past Google trends data, and combinations of the three. Individual results from DJIA, TSX, JSE and SENSEX all suggest that the DDPG algorithm performs best when using past returns and trading volume as predictors. Also, the combination of returns, trading volume and Google search volume yields promising results, especially for JSE and SENSEX. In finance, past returns and trading volume are historically important predictors of future returns. Google search volume is a relatively new predictor, only having been available for about a decade. Therefore, we focus our analysis on the R;TV and R;TV;G agents, and present detailed results for these agents across all four markets. Comprehensive results for all agents on DJIA, TSX, JSE and SENSEX can be found in appendix A.3.

First, we test the DDPG algorithm without transaction cost. The model is not restricted in any way. It is allowed to change portfolio composition daily, which it usually does. In fact, the average length of a trade is only about 1.3 days for these unrestricted agents. Additionally, the agents generally only hold one stock at the time, see section 5.3 for the analysis of algorithm behavior.

Table 5.1: Annualized performance metrics for DDPG agents on each stock market, along with B&H portfolios, over the testing period 2008-2019. Agents are allowed to trade on a daily basis, and therefore predict daily returns. $\tau = 0.00\%$.

	DJI			TSX			JSE			SENSEX		
	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H
Mean Returns [%]	24.04	13.19	10.53	14.17	2.96	8.73	26.73	30.79	16.77	28.83	25.81	23.42
Volatility [%]	28.55	25.61	18.01	30.85	33.15	14.97	33.53	32.36	18.48	37.36	33.69	23.06
Sharpe Ratio	0.84	0.52	0.58	0.46	0.09	0.58	0.80	0.95	0.91	0.77	0.77	1.02
FF-5 alpha [%]	19.20***	7.37	3.08	17.26	8.19	7.72	29.13	32.04*	15.71	32.40	27.30	21.97

Note: *** $p < .01$, * $p < .1$ statistically different from B&H

Table 5.1 compares the performance of the DDPG algorithm using either return and trading volume, or return, trading volume and Google trends search data as predictors. The agents at-

tempt to maximize return over the testing period. On average, over all markets, the R;TV DDPG agents obtain a mean return 8.58 percentage points (p.p.) higher than the B&H benchmark. The DDPG portfolios are quite a lot more volatile than the benchmarks, and this decreases risk-adjusted returns. In spite of this, the R;TV agent on DJIA is able to achieve a higher Sharpe ratio than its benchmark. Moreover, this agent also attains a positive Fama-French intercept which is statistically higher than the benchmark portfolio.

The daily Google search volume predictor does not seem to add value on the DJIA or the TSX markets. Performance deteriorates both in terms of mean returns and risk-adjusted returns. However, search volume seems more valuable for JSE and SENSEX. On JSE, the R;TV;G agent actually outperforms B&H by 14.02 p.p. in terms of mean return. In fact, the JSE R;TV;G alpha is statistically different from the B&H benchmark. The agent also obtains a higher Sharpe ratio than the benchmark. Russell (2017) classifies South Africa and India as emerging markets. The United States and Canada are, on the other hand, developed markets. This difference in market efficiency could offer some explanation as to why Google searches look more valuable as a predictor on JSE and SENSEX.

The algorithm is able to achieve high returns, beating the benchmark, in all markets over the whole testing period. Therefore it is interesting to investigate its practical potential by accounting for transaction cost in the implementation. If the agents are allowed to trade daily, completely updating their portfolios at each time step, portfolio returns are dominated by transaction cost. To deal with this issue we restrict the agent to only trade once a week, once a month or once a quarter. A weekly trading frequency is quickly deemed unviable, these agents obtain a negative return over the testing period in all markets except SENSEX. Consequently, results from the agents trading once a week are not discussed further.

Table 5.2: Annualized performance metrics for DDPG agents on each stock market, along with B&H portfolios, over the testing period 2008-2019. Agents are restricted to only trade once every month, and therefore predict monthly returns. $\tau = 0.25\%$.

	DJI			TSX			JSE			SENSEX		
	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H
Mean Returns [%]	11.02	8.39	10.53	15.50	2.07	8.73	16.59	4.83	16.77	23.63	25.28	23.42
Volatility [%]	27.17	27.96	18.01	38.23	36.76	14.97	33.83	32.15	18.48	38.16	40.46	23.06
Sharpe Ratio	0.41	0.30	0.58	0.41	0.06	0.58	0.49	0.15	0.91	0.62	0.62	1.02
FF-5 alpha [%]	6.37	3.86	3.08	21.15	8.32	7.72	18.62	7.02	15.71	25.23	28.99	21.97

The results for the agent restricted to only trade once per month are summarized in table 5.2. Interestingly, R;TV stills seems to be the most promising combination of predictors, like before transaction cost was imposed. These agents beat their respective benchmarks by 1.82 p.p. on average across all markets. Google search volume only looks to add predictive value for SENSEX, where the R;TV;G agent obtains a 1.86 p.p. higher return than the B&H portfolio.

Table 5.3: Annualized performance metrics for DDPG agents on each stock market, along with B&H portfolios, over the testing period 2008-2019. Agents are restricted to only trade once every quarter, and therefore predict quarterly returns. $\tau = 0.25\%$.

	DJI			TSX			JSE			SENSEX		
	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H	R;TV	R;TV;G	B&H
Mean Returns [%]	12.06	2.38	10.53	18.64	2.91	8.73	20.46	18.24	16.77	17.08	27.23	23.42
Volatility [%]	27.31	29.44	18.01	37.17	28.37	14.97	36.27	31.81	18.48	39.45	43.67	23.06
Sharpe Ratio	0.44	0.08	0.58	0.50	0.10	0.58	0.56	0.57	0.91	0.43	0.62	1.02
FF-5 alpha [%]	7.00	-1.95	3.08	24.05*	4.75	7.72	24.44	19.70	15.71	23.30	31.76	21.97

Note: * $p < .1$ statistically different from B&H

The results for the agent restricted to quarterly trading are displayed in table 5.3. Also here, using R;TV looks to be the best set of predictors for return. The R;TV agents beat B&H by 2.2 p.p. on average across all markets in terms of mean return. Notably, in the TSX market, the agent actually achieves a positive alpha that is significantly different from that of the buy-and-hold. SENSEX is the only market where R;TV is outperformed by the benchmark. Interestingly, adding Google search words for the SENSEX agent improves the mean return to surpass the benchmark by 3.81 p.p. Google trends also shows promise is JSE, where the agent beats the B&H by 1.47 p.p. in terms of mean returns. This is in accordance with the hypothesis that Google searches is a more useful predictor in somewhat less developed markets.

The restricted agents exhibit similar behavior to their non-restricted equivalents in that they on average hold one stock at the time. Therefore, volatilities are largely unchanged compared to table 5.1. As expected, mean return is generally lower after transaction cost is imposed compared to before. This is true for both monthly and quarterly trading horizons, TSX, with a quarterly horizon, being the only exception. The explanation is likely two-fold. First of all, the direct cost of making a transaction decreases portfolio values. Secondly, the results suggest the agents are better at predicting daily returns than monthly or quarterly. Generally, our findings indicate a trading horizon between one month and one quarter is the best approach for handling transaction cost.

We further assess the DDPG performance by comparing it to that of a simple linear panel data regression on both the monthly and quarterly trading horizon, see chapter 4 for implementation details.

Table 5.4: Annualized mean return for DDPG agents and the linear regressions over the testing period 2008-2019. Agents are restricted to trade only once every month or once every quarter. Agents predict monthly or quarterly returns respectively. $\tau = 0.25\%$.

		DJIA		TSX		JSE		SENSEX	
		R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G
Monthly	DDPG	11.02	8.39	15.50	2.07	16.59	4.83	23.63	25.28
	Lin.Reg	8.88	4.97	-1.03	-0.59	6.80	3.44	3.86	-0.53
Quarterly	DDPG	12.08	2.39	18.68	2.91	20.50	18.15	17.18	27.39
	Lin.Reg	7.45	2.24	13.61	-2.11	11.44	11.15	1.67	2.44

Table 5.4 lists the annualized mean return for both the monthly and the quarterly DDPG agents and for the linear regression. We see that our agents outperform the equivalent linear regressions across all markets, although to a varying degree. The results suggest that the DDPG algorithm is able to extract more information, possibly by identifying more complex relationships, out of the predictors compared to the simple linear regression model. Performance difference between DDPG and the regression is close to constant for the two considered prediction horizons. The monthly DDPG agents outperform the regression by 10.2 p.p. on average across all markets, compared to 8.92 p.p. for the quarterly agents.

To summarize, DDPG agents are able to beat their relevant benchmark both before and after transaction cost is imposed. Positive alphas, significantly differing from the equivalent B&H intercepts, are achieved on DJIA, JSE and TSX. Including returns and trading volume as predictors looks to be valuable when using the DDPG algorithm to predict stock returns. Additionally, Google search volume may contribute positively to the prediction of stock returns in developing markets. Finally, the results suggest that the algorithm is best at predicting daily returns, with horizons ranging between a month and a quarter being adequate alternatives when transaction cost is imposed.

5.3 DDPG Trading Behavior

A known limitation of deep learning methods is a low model interpretability. It is generally difficult to inspect how the algorithm achieves its results. In the following section, we analyze the algorithm behavior in an attempt to shed light on what happens "under the hood" of this black-box algorithm. We do so through a thorough analysis of results on the DJIA. The model exhibits similar behavior across all markets. Additionally, we look at the consequences for algorithm behavior when transaction cost is imposed. Finally, we review whether handling transaction cost by restricting trading frequency is optimal through looking at potential unintended consequences of the solution.

Table 5.5: Daily performance metrics for DDPG agents using past return, trading volume and Google search volumes as predictors on the DJIA market, along with a buy-and-hold strategy. Agents are allowed to trade on a daily basis, and therefore predict daily returns. Testing period: 2008-2019. $\tau = 0.00\%$.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	0.05	0.03	0.00	0.10	0.02	0.04	0.05	0.04
Std. Dev. [%]	1.70	1.82	1.60	1.80	1.67	1.76	1.61	1.13
Skewness	-0.50	-0.49	-0.34	1.37	-0.31	-0.04	0.26	-0.05
Kurtosis	13.51	17.02	8.61	19.00	7.20	15.42	8.64	10.36
$VaR_{5\%}$ [%]	2.51	2.58	2.42	2.53	2.56	2.56	2.41	1.72
$VaR_{95\%}$ [%]	2.41	2.54	2.28	2.67	2.50	2.49	2.41	1.56
$CVaR_{5\%}$ [%]	4.02	4.43	4.03	3.87	4.08	4.12	3.79	2.80
$CVaR_{95\%}$ [%]	4.05	4.10	3.61	4.57	3.82	4.28	3.81	2.62
Hit Ratio	0.52	0.51	0.51	0.53	0.51	0.51	0.50	0.55
Avg. Returns+ [%]	1.11	1.15	1.00	1.15	1.10	1.14	1.12	0.69
Avg. Returns- [%]	-1.14	-1.20	-1.10	-1.13	-1.15	-1.14	-1.11	-0.77

Table 5.5 displays various performance metrics for DDPG agents utilizing different sets of predictors on the DJIA. The agents can, and do, trade on a daily basis without the occurrence of transaction cost. The table implies that all agents prefer to maximize return over minimizing risk. B&H clearly has a lower daily volatility than all agents. In fact, the best performing DDPG agent, R;TV, has 0.63 p.p. higher daily standard deviation compared to the benchmark. The values of $5\%VaR$ and $5\%CVaR$ also indicate a preference for high returns over low risk, with the average $5\%VaR$ for all DDPG agents being 0.79 p.p. higher than B&H. However, these findings are not overly surprising considering that the objective of the algorithm is to optimize log returns, and not a risk-adjusted metric such as Sharpe ratio, see chapter 4.

Interestingly, the R;TV and the R;TV;G agents, which are the two strategies with the highest daily mean returns, are also the only ones with a positively skewed returns distribution, in addition to B&H. The positive skew tells us that the agents generate a substantial number of extreme positive returns, increasing the daily mean return. It seems the high DDPG returns come from fewer large positive returns rather than more frequent smaller gains. This theory is supported by the fact that all DDPG agents have higher $95\%VaR$ and $95\%CVaR$ than B&H. Also, compared to B&H, the DDPG agents on average have a 6.75% lower hit ratio, further supporting the hypothesis. Moreover, we see that the Avg. Returns+ metric is higher across all DDPG agents compared to B&H, while the B&H strategy has more desirable Avg. Returns-; again it seems DDPG mean returns rely on more extreme values.

The DDPG agents' preference for risky investments is found across all markets, see appendix A.3. Table 5.6 may help to explain what causes the risky behavior. On average, the algorithm

switches position more than once every two days. Additionally, at any time step, it usually chooses to hold a single stock in its portfolio; risk is not diversified.

Table 5.6: Trading characteristics for DDPG R;TV for each market. A trade starts at the moment the agent puts more than 1% of its money in a stock, and lasts until the position is liquidated. $\tau = 0.25\%$

	DJIA	TSX	JSE	SENSEX
Number of trades	2542	2725	3810	2078
Average trade duration [Days]	1.32	1.29	1.15	1.47

Table 5.7 lists annualized performance metrics for the DDPG agents, along with the B&H portfolio.

Table 5.7: Annualized performance metrics for DDPG agents using past return, trading volume and Google search volumes as predictors on the DJIA market, along with a Buy-and-Hold strategy. Testing period: 2008-2019. $\tau = 0.25\%$.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	13.70	7.79	0.20	24.04	5.62	8.82	13.19	10.53
Std. Dev. [%]	26.94	28.86	25.33	28.55	26.45	27.94	25.61	18.01
Sharpe Ratio	0.51	0.27	0.01	0.84	0.21	0.32	0.52	0.58
FF-5 alpha [%]	8.22	2.99	-6.41	19.20***	-0.73	4.22	7.37	3.08

Note: *** $p < .01$ statistically different from B&H
Dataset: Fama/French 5 Factors [Daily]

The fact that the DDPG agents generally only hold one stock at the time makes it all the more impressive that they are able to compete with the B&H Sharpe ratios, as we can see from table 5.7.

The average time the agent holds a position indicates that transaction cost could a problem for the framework. The following graph shows a plot of the DDPG agents' portfolio value after implementing transaction cost.

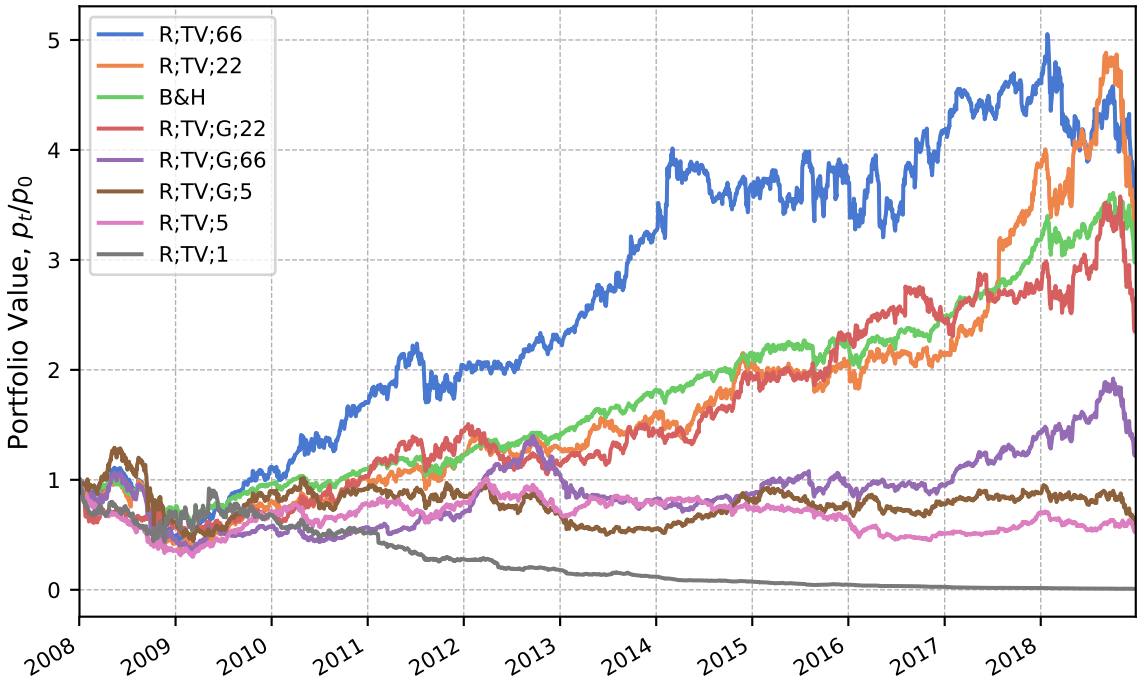


Figure 5.1: Cumulative return of DDPG and buy-and-hold strategy on DJIA over the time period 2008-2019. Agents are restricted to only trade once every week, month or quarter. Thus. they predict weekly, monthly or quarterly returns respectively. $\tau = 0.25\%$. Legend sorted by descending cumulative return.

In figure 5.1, (R;TV;1) shows the R;TV agent trading on a daily basis when transaction cost is charged. The portfolio tends toward 0, since algorithm behavior is largely unchanged. The same pattern is observed for all agents trading every day, across all markets. As shown in section 5.2, restricting how often the agent can trade could be a feasible solution to the problem, at least in terms of beating the B&H benchmark. Somewhere between one month and one quarter seems to be the most usable trading horizon for the DDPG algorithm. Trading once a week seems too frequent, a statement supported by figure 5.1. This is true also for the remaining markets, see appendix A.3. Also discussed in section 5.2, performance decreases when restricting the trading frequency, with TSX being the only exception. We hypothesize that, in addition to the direct impact of transaction cost, the performance deterioration could be explained by the DDPG algorithm finding it more challenging to predict monthly or quarterly returns compared to predicting daily returns.

A subtle issue with restricting the trading frequency is that many data points in the testing set are lost, and signals the agent normally would act upon are lost along with them. The optimal solution to tackling transaction cost could perhaps be to only let the algorithm trade if it predicted that, taking transaction cost into consideration, the trade would be profitable. Then, no data points in the testing set would be lost, and all profitable opportunities would be available for the agent to exploit. Unfortunately, this is an impractical approach since predictions

are not explicitly observed in the framework.

In an attempt to further explain how the agents achieve high mean returns we plot the price development of the average stock the model holds up until it makes a trade, along with the price development of said stock and the average stock it buys in the trade, replacing the old stock (figure 5.2. We average over all markets, for monthly and quarterly R;TV agents.

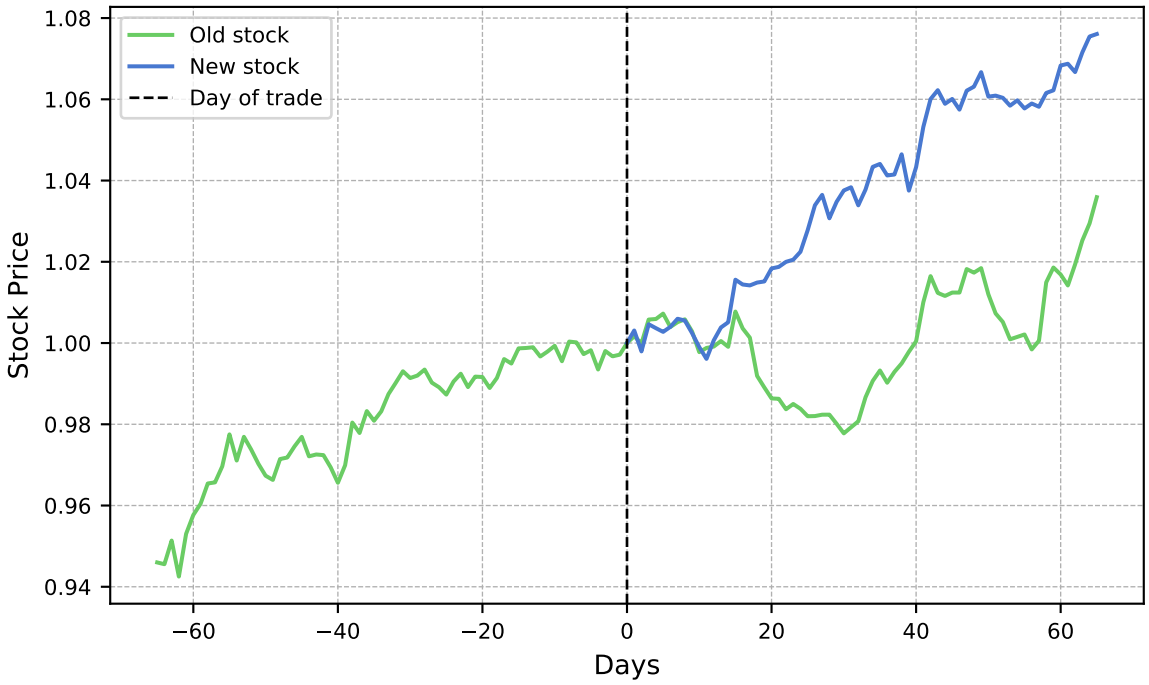


Figure 5.2: Left: average price development of stock held before day of trade. Right: price development of stock that is bought, New stock, and the stock no longer held, Old stock, after the trade. The graphs are based on averages from all markets, specifically the R;TV agents trading once a month or once a quarter.

Interestingly, on average across all markets, the DDPG agents are able to replace a stock with a better performing one when it does a trade. This seems to be a trait of the algorithm's behavior. Based on these results, the algorithm is likely to perform better when allowed to switch positions as long as the trade is profitable after transaction cost. With the current solution of restricting trading frequency, if the agent enters a bad position it is unable to leave until a month or a quarter has passed - and bad positions are more likely to occur when relying only on a single stock.

To summarize the behavioral study of the DDPG algorithm, the trading strategy of all the agents is quite risky and results are driven by a few, large gains. If left unrestricted, the agents prefer to switch positions frequently. Generally, at any time step, the agents prefer to keep a single stock in their portfolio. This explains the extreme returns, but also the volatile nature of the resulting strategy. In a more realistic situation, the frequent switching of positions leaves the

agent with unbearable transaction costs. Handling transaction cost by restricting the trading frequency of the agents is viable, as many agents beat their relevant benchmark. However, the solution has the unintended consequence of limiting the agents' ability to improve their position in the market.

6 Conclusion

Google DeepMind shook the AI landscape by achieving extreme results playing the Atari 2600 games, and later the game of Go and Chess, using deep reinforcement learning. These findings motivated the exploration of possible applications outside of game playing. In this study we evaluate the performance of a reinforcement learning algorithm, the Deep Deterministic Policy Gradient, by trading stocks included in stock indices in four countries: the United States, Canada, South Africa and India.

The algorithm is tested using past log returns (R), past trading volume (TV), Google search volume (G), and combinations of the three as predictors. We find that the combination of returns and trading volume lead to the best performing agents. Also, Google search volume seem to add more value as predictors in emerging markets (South Africa and India) compared to developed markets (USA and Canada).

The R;TV agent, on average across all markets, outperforms the buy-and-hold benchmark by 8.58 percentage points (p.p.) in terms of annualized mean return. This corresponds to a difference of 93.53 p.p. over the 2008-2019 testing period. The R;TV agent on DJIA attains positive Fama-French Five-Factor intercept which is statistically higher than the equivalent buy-and-hold alpha.

The daily Google search volume predictor does not seem to add any value in the DJIA or the TSX markets. For JSE, however, the R;TV;G agent obtains an annualized mean return slightly higher than its R;TV counterpart. Moreover, in addition to obtaining a higher annualized mean return, the R;TV;G agent surpasses the B&H benchmark in terms of Sharpe ratio. The agent yields a Fama-French Five-Factor intercept statistically higher than the benchmark.

We also test the DDPG algorithm using the same predictors on the same markets after accounting for transaction cost. Three types of agents are evaluated: one is restricted to trade only once a week, one to trade only once a month, the final to trade only once a quarter. In terms of annualized mean return, the DDPG agents allowed to trade once a month outperform the buy-and-hold benchmarks by an average of 1.82 p.p. across all markets. The equivalent agents allowed to trade once a quarter outperform the buy-and-hold benchmarks by an average of 2.2 p.p. Using a trading horizon of between one month and one quarter looks to yield the best results. Google search volume seems to add predictive value for SENSEX, where the R;TV;G agent obtains a 1.86 p.p. higher annualized mean return than the benchmark.

We benchmark the DDPG algorithm with transaction cost further by comparing results to those of a simple linear regression. The agents trading once a month outperform the regression by 10.2 p.p. on average across markets, compared to 8.92 p.p. for the agents trading once a quarter.

Applying the DDPG algorithm to stock trading is largely novel work. Naturally, there are several exciting areas of further research. Most notably, experimenting with different reward functions that decrease risk appetite would be interesting. Another could be to find ways to make the algorithm diversify its portfolio.

Bibliography

Aalborg, H. A., Molnár, P., and de Vries, J. E. (2018). What can explain the price, volatility and trading volume of Bitcoin? *Finance Research Letters*.

Allen, F. and Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*.

Aouadi, A., Arouri, M., and Teulon, F. (2013). Investor attention and stock market activity: Evidence from France. *Economic Modelling*, 35:674–681.

Arévalo, A., Niño, J., Hernández, G., and Sandoval, J. (2016). High-Frequency Trading Strategy Based on Deep Neural Networks. In *Intelligent Computing Methodologies*. Springer International Publishing.

Arifovic, J. (1994). Genetic algorithm learning and the cobweb model. *Journal of Economic Dynamics and Control*.

Arifovic, J. (2001). Evolutionary dynamics of currency substitution. *Journal of Economic Dynamics and Control*.

Bank, M., Larch, M., and Peter, G. (2011). Google search volume and its influence on liquidity and returns of German stocks. *Financial markets and portfolio management*, 25(3):239.

Bijl, L., Kringhaug, G., Molnár, P., and Sandvik, E. (2016). Google searches and stock returns. *International Review of Financial Analysis*, pages 150–156.

Campbell, J. Y., Grossman, S. J., and Wang, J. (1992). Trading Volume and Serial Correlation in Stock Returns. Technical report, National Bureau of Economic Researcg.

Carneiro, H. A. and Mylonakis, E. (2009). Google trends: a web-based tool for real-time surveillance of disease outbreaks. *Clinical infectious diseases*, 49(10):1557–1564.

Challet, D. and Ayed, A. B. H. (2013). Predicting financial markets with Google Trends and not so random keywords. *arXiv preprint arXiv:1307.4643*.

Chang, T.-J., Yang, S.-C., and Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*.

Chen, J., Tang, G., Yao, J., and Zhou, G. (2018). Investor Attention and Stock Returns. *SSRN*.

Choi, H. and Varian, H. (2012). Predicting the Present with Google Trends. *The Economic Record*, pages 2–9.

Chong, E., Han, C., and Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*.

Chourmouziadis, K. and Chatzoglou, P. D. (2016). An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Systems with Applications*.

Cumming, J. (2015). An Investigation into the Use of Reinforcement Learning Techniques within the Algorithmic Trading Domain. Master's thesis, Imperial College London.

Da, Z., Engelberg, J., and Gao, P. (2011). In Search of Attention. *The Journal of Finance*, 56(5).

Da, Z., Engelberg, J., and Gao, P. (2014). The sum of all FEARS investor sentiment and asset prices. *The Review of Financial Studies*, 28(1):1–32.

Dempster, M. and Jones, C. M. (2001). A real-time adaptive trading system using Genetic Programming. *Quantitative Finance*, 1:397–413.

Dempster, M. and Leemans, V. (2006). Learning to Trade via Direct Reinforcement. *Expert Systems With Applications*, 30(3):543.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017). Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *Transactions on Neural Networks and Learning Systems*, 28(3):653–664.

Dimpfl, T. and Jank, S. (2016). Can internet search queries help to predict stock market volatility? *European Financial Management*, 22(2):171–192.

Ding, X., Zhang, Y., Liu, T., and Duan, J. (2015). Deep Learning for Event-driven Stock Prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press.

Emami, P. (2016). Deep Deterministic Policy Gradients in TensorFlow. `https://pemami4911.github.io/blog/2016/08/21/ddpg-rl.html#References`. Accessed: 2018-10-29.

Eysenbach, G. (2006). Infodemiology: tracking flu-related searches on the web for syndromic surveillance. In *AMIA Annual Symposium Proceedings*, volume 2006, page 244. American Medical Informatics Association.

Fama, E. F. and French, K. R. (2014). A Five-Factor Asset Pricing Model. *Journal of Financial Economics*.

Filos, A. (2018). Reinforcement Learning for Portfolio Management. Master's thesis, Imperial College London.

Financial Stability Board (2017). *Artificial intelligence and machine learning in financial services Market developments and financial stability implications*. FSB.

Fink, C. and Johann, T. (2013). May I Have Your Attention, Please: The Market Microstructure of Investor Attention. *SSRN Electronic Journal*.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.

Freitas, F. D., Souza, A. F. D., and de Almeida, A. R. (2009). Prediction-based portfolio optimization model using neural networks. *Neuroscience*, 72:2155–2170.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., and Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012.

Goddard, J., Kita, A., and Wang, Q. (2015). Investor attention and FX market volatility. *Journal of International Financial Markets, Institutions and Money*, 38:79–96.

Gwilym, O. A., Hasan, I., Wang, Q., and Xie, R. (2016). In Search of Concepts: The Effects of Speculative Demand on Stock Returns. *European Financial Management*, 22(3):427–449.

Harford, T. (2017). Just google it: The student project that changed the world. *Accesible online http://www. bbc. com/news/business-39129619*.

Heaton, J., Polson, N., and Witte, J. (2016). Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*.

Hu, H., Tang, L., Zhang, S., and Wang, H. (2018). Predicting the direction of stock markets using optimized neural networks with Google Trends. *Neurocomputing*, 285:188–195.

Huang, C. Y. (2018). Financial Trading as a Game: A Deep Reinforcement Learning Approach. *arXiv preprint arXiv:1807.02787*.

Islam, R., Henderson, P., Gomrokchi, M., and Precup, D. (2017). Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control. *Transportation Quarterly*.

Jevne, H. K., Haddow, P. C., and Gaivoronski, A. A. (2012). Evolving constrained mean-var efficient frontiers. In *2012 IEEE Congress on Evolutionary Computation*.

Jiang, Z., Xu, D., and Liang, J. (2017). A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. *arXiv preprint arXiv:1706.10059*.

Joseph, K., Wintoki, M. B., and Zhang, Z. (2011). Forecasting abnormal stock returns and trading volume using investor sentiment: Evidence from online search. *International Journal of Forecasting*, 27(4):1116–1127.

Kahneman, D. (2013). *Attention and Effort*. PRENTICE-HALL INC, Englewood Cliffs, New Jersey.

Karpoff, J. M. (1987). The Relation Between Price Changes and Trading Volume: A Survey. *Journal of Financial and Quantitative Analysis*.

Kim, K.-J. and Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with applications*.

Kim, N., Lučivjanská, K., Molnár, P., and Villa, R. (2018). Google searches and stock market activity: Evidence from Norway. *Finance Research Letters*, pages 35–52.

Kirkpatrick II, C. D. and Dahlquist, J. A. (2010). *Technical analysis: the complete resource for financial market technicians*. FT press.

Krink, T. and Paterlini, S. (2011). Multiobjective optimization using differential evolution for real-world portfolio optimization. *Computational Management Science*, 8:157–179.

Kristoufek, L. (2013). Can Google Trends search queries contribute to risk diversification? *Scientific reports*, 3:2713.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 2*, pages 1097–1105. Curran Associates, Inc.

Liang, Z., Chen, H., Zhu, J., Jiang, K., and Li, Y. (2018). Deep Reinforcement Learning in Portfolio Management. *arXiv preprint arXiv:1808.09940*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous Control with Deep Reinforcement Learning. *arXiv preprint arXiv:1509.02971*.

Lin, C.-C. and Liu, Y.-T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, 185:393–404.

Mahfoud, S. and Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*.

Mauro, G. (2016). Six graphs to understand the state of Artificial Intelligence academic research. https://blog.ai-academy.com/six-graphs-to-understand-the-state-of-ai-academic-research-3a79cac4c9c2. Accessed: 2018-12-6.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., len King, H., Kumaran, D., Wierstra, D., Legg, S., and Hass-abis, D. (2015). Human-level control through deep reinforcement learning. *Nature*.

Molnár, P. and Bašta, M. (2017). Google searches and Gasoline prices. In *2017 14th International Conference on the European Energy Market (EEM)*, pages 1–5.

Moody, J. and Saffell, M. (2001). Learning to Trade via Direct Reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889.

Neely, C., Weller, P. A., and Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? a genetic programming approach. Working Papers 1996-006, Federal Reserve Bank of St. Louis.

Neely, C. J. and Weller, P. A. (1999). Technical trading rules in the European Monetary System. *Journal of International Money and Finance*.

Niaki, S. T. A. and Hoseinzade, S. (2013). Forecasting S&P500 index using artificial neural networks and design of experiments. *Journal of Industrial Engineering International*.

Oshihara, A., Fujikawa, K., Seki, K., and Uehara, K. (2014). Predicting Stock Market Trends by Recurrent Deep Neural Networks. In *RICAI 2014: Trends in Artificial Intelligence*. Springer International Publishing.

Pelat, C., Turbelin, C., Bar-Hen, A., Flahault, A., and Valleron, A.-J. (2009). More diseases tracked by using Google Trends. *Emerging infectious diseases*, 15(8):1327.

Polgreen, P. M., Chen, Y., Pennock, D. M., Nelson, F. D., and Weinstein, R. A. (2008). Using internet searches for influenza surveillance. *Clinical infectious diseases*, 47(11):1443–1448.

Preis, T., Moat, H. S., and Stanley, H. E. (2013). Quantifying trading behavior in financial markets using Google Trends. *Scientific reports*, 3:1684.

Preis, T., Reith, D., and Stanley, H. E. (2010). Complex dynamics of our economic life on different scales: insights from search engine query data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1933):5707–5719.

Russell, F. (2017). FTSE Equity Country Classification Process. `https://research.ftserussell.com/products/downloads/FTSE_Equity_Country_Classification_Paper.pdf`. Accessed: 2019-10-6.

Shin, K.-S. and Lee, Y.-J. (2002). A genetic algorithm application in bankruptcy prediction modeling. *Expert Systems with Applications*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic Policy Gradient Algorithms. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages I–387–I–395. JMLR.org.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G.,

Graepel, T., and Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*.

Soleimani, H., Golmakani, H. R., and Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Syst. Appl.*, 36:5058–5063.

Sutton, R. and Barto, A. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Takeda, F. and Wakao, T. (2014). Google search intensity and its relationship with returns and trading volume of Japanese stocks. *Pacific-Basin Finance Journal*, 27:1–18.

Tantaopas, P., Padungsaksawasdi, C., and Treepongkaruna, S. (2016). Attention effect via internet search intensity in Asia-Pacific stock markets. *Pacific-Basin Finance Journal*, 38:107–124.

Tsao, C.-Y. (2010). Portfolio selection based on the mean–VaR efficient frontier. *Quantitative Finance*.

Tsao, C.-Y. and Liu, C.-K. (2006). Incorporating Value-at-Risk in Portfolio Selection: An Evolutionary Approach. In *9th Joint International Conference on Information Sciences (JCIS-06)*. Atlantis Press.

Vlastakis, N. and Markellos, R. N. (2012). Information demand and stock market volatility. *Journal of Banking & Finance*, 36(6):1808–1821.

Vozlyublennaia, N. (2014). Investor attention, index performance, and return predictability. *Journal of Banking & Finance*, 41:17–35.

Wang, J. (2000). Trading and hedging in S&P 500 spot and futures markets using genetic programming. *Journal of Futures Markets*, 20:911 – 942.

Wang, Z., Qian, Y., and Wang, S. (2018). Dynamic trading volume and stock return relation: Does it hold out of sample? *International Review of Financial Analysis*.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neuroscience*, 50:159–175.

Appendix

A | Appendix

A.1 DDPG Hyperparameters

Table A.1 displays the hyperparameters of the DDPG algorithm used during training.

Table A.1: Hyperparameters for DDPG.

Hyperparameters	Value	Description
Batch size	64	Mini batch during training
Epochs	300	Number of training epochs
Warm up size	10^4	Minimum size of replay buffer before learning
Replay buffer size	10^6	Size of replay buffer
Grad norm clip	1.24	Maximum allowed gradient normalization
Noise cap	0.15	Maximum size of added noise
Target update interval	3	How often target networks are updated
$\tilde{\sigma}$	0.085	Standard deviation of smoothing regularization noise
σ	0.065	Standard deviation of exploration noise
γ	0.93	Reward discounting factor
τ	0.006	Target network update ratio
α	14×10^{-3}	Actor learning rate
β	18×10^{-3}	Critic learning rate
Hidden 1	400	Number of nodes in first hidden layer
Hidden 2	100	Number of nodes in second hidden layer
Hidden 3	60	Number of nodes in third hidden layer

A.2 Google Search Words

Table A.2 shows, for each market, the search words used to collect Google search volume data.

Table A.2: Associated Google trends search word for each market's component stock. Filtered on their respective countries searches and downloaded for time period 01/01/2004 - 12/31/2018.

Google Search Words
DJIA
3M, American Express, Apple, Boeing, Caterpillar, Chevron, Cisco, Coca-Cola, Walt Disney, Exxon Mobil, Goldman Sachs, Home Depot, IBM, Intel, Johnson & Johnson, JPMorgan Chase, Mcdonald's, Merck, Microsoft, Nike, Pfizer, Procter & Gamble, TRV, UTX, UnitedHealth, Verizon, Walmart, WBA
TSX
Toronto Dominion Bank, Bank of Nova Scotia, Enbridge, Canadian National Railway, Bank of Montreal, Canadian Imperial Bank of Commerce, Canadian Natural Resources, Manulife, BCE, Brookfield, TransCanada, Canadian Pacific Railway, Imperial Oil, Sun Life Financial, Rogers Communications, Telus, Loblaw, WCN, National Bank of Canada, Pembina Pipeline, Teck, Barrick Gold, CGI Group, Husky Energy, Fortis, Saputo, Encana, Power Corporation of Canada, George Weston, AEM
JSE
Absa Group, Anglo American, AngloGold Ashanti, Aspen Pharmacare, BHP Group, British American Tobacco, Bidvest, CFR, CLS, Capitec Bank, Discovery, FirstRand, Gold Fields, Life Healthcare, Mr Price, MTN, Nedbank, Naspers, Netcare, Remgro, RMB, Sappi, Standard Bank Group, Shoprite, Sanlam, Sasol, Tiger Brands, Foschini Group, Truworths, Vodacom, Woolworths
SENSEX
Asian Paints, Axis Bank, Bajaj Finance, Bharti Airtel, HDFC Bank, HCL Technologies, Hero Honda, Hindustan Copper, HDFC, ICICI Bank, IndusInd Bank, ITC, Kotak Mahindra Bank, Larsen & Toubro, Mahindra & Mahindra, Maruti Suzuki India, ONGC, Reliance Industries, State Bank of India, Sun Pharmaceutical, Tata Consultancy Services, Tata Motors, Tata Steel, Vedanta

A.3 Detailed Metrics for DDPG on all Markets

Comprehensive results for the DDPG algorithm on each market are displayed in this section. The results include figures of cumulative returns (A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8), tables of daily performance metrics (A.3, A.5, A.7, A.9, A.11, A.13, A.15) and tables of annualized performance metrics (A.4, A.6, A.8, A.10, A.12, A.14, A.16), both before and after transaction cost is imposed[1].

A.3.1 Dow Jones Industrial Average

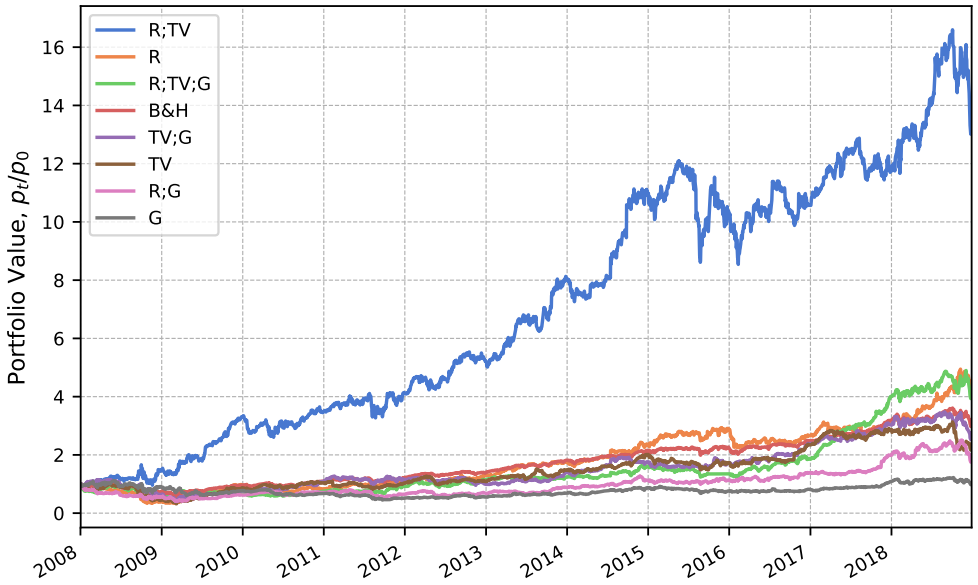


Figure A.1: Cumulative return of DDPG and buy-and-hold strategy on DJIA over the time period 2008-2019. Agents trade on a daily basis and predict daily returns. $\tau = 0.00\%$. Legend sorted by descending cumulative return.

[1]Daily and annualized performance metrics for DJIA after transaction cost can be found in section 5.3

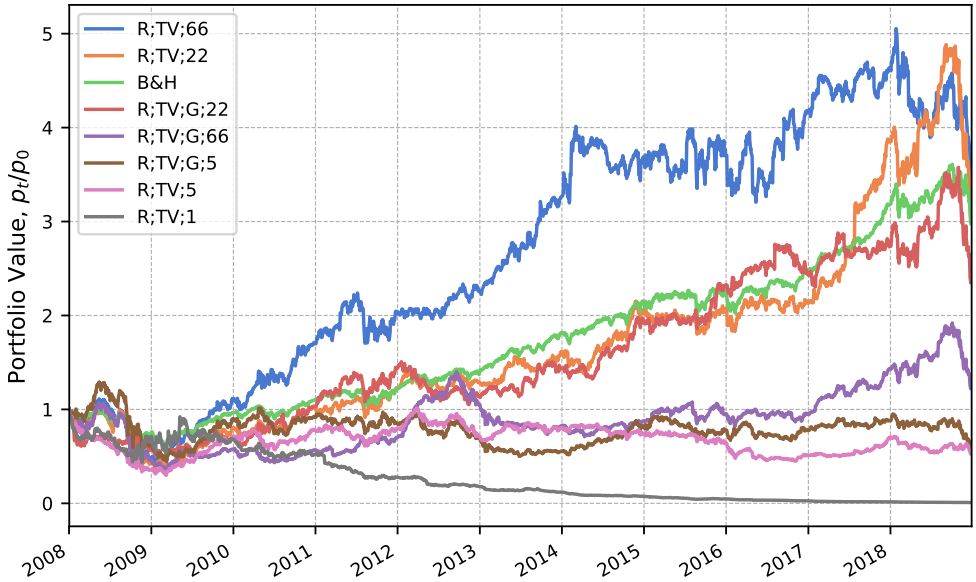


Figure A.2: Cumulative return of DDPG and buy-and-hold strategy on DJIA over the time period 2008-2019. Agents are restricted to only trade once every week, month or quarter. Thus. they predict weekly, monthly or quarterly returns respectively. $\tau = 0.25\%$. Legend sorted by descending cumulative return.

Table A.3: Daily metrics for DDPG run with trading horizon and buy-and-hold strategy on DJIA. Trading horizons are either 1, 5, 22, or 66 days. $\tau = 0.25\%$.

	1		5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	-0.17	-0.12	-0.02	-0.02	0.04	0.03	0.05	0.01	0.04
Std. Dev. [%]	2.02	2.00	1.73	1.92	1.71	1.76	1.72	1.86	1.13
Skewness	0.73	1.74	0.07	0.23	-0.52	0.02	-0.47	-0.54	-0.05
Kurtosis	17.13	29.38	7.51	16.53	12.21	6.71	11.49	12.64	10.36
$VaR_{5\%}$ [%]	2.94	2.89	2.74	2.88	2.62	2.68	2.55	2.79	1.72
$VaR_{95\%}$ [%]	2.46	2.47	2.58	2.67	2.62	2.63	2.59	2.56	1.56
$CVaR_{5\%}$ [%]	4.93	4.64	4.25	4.54	4.18	4.24	4.19	4.61	2.80
$CVaR*_{95\%}$ [%]	4.71	4.68	3.91	4.31	3.97	4.22	4.07	4.17	2.62
Hit Ratio	0.43	0.44	0.50	0.49	0.52	0.51	0.52	0.52	0.55
Avg. Returns+ [%]	1.26	1.26	1.16	1.25	1.12	1.18	1.14	1.15	0.69
Avg. Returns- [%]	-1.28	-1.23	-1.22	-1.27	-1.16	-1.19	-1.14	-1.28	-0.77

Table A.4: Annualized metrics for DDPG run with trading horizon and buy-and-hold strategy on DJIA. Trading horizons are either 0, 5, 22, or 66 days. $\tau = 0.25\%$.

	1		5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	-44.09	-29.43	-5.37	-4.30	11.02	8.39	12.08	2.39	10.53
Std. Dev. [%]	32.01	31.71	27.45	30.42	27.17	27.96	27.37	29.47	18.01
Sharpe Ratio	-1.38	-0.93	-0.20	-0.14	0.41	0.30	0.44	0.08	0.58
FF-5 alpha [%]	-47.81	-34.47	-12.05	-8.68	6.37	3.86	7.06	-1.92	3.08

Dataset: Fama/French 5 Factors [Daily]

A.3.2 S&P/Toronto Stock Exchange Index

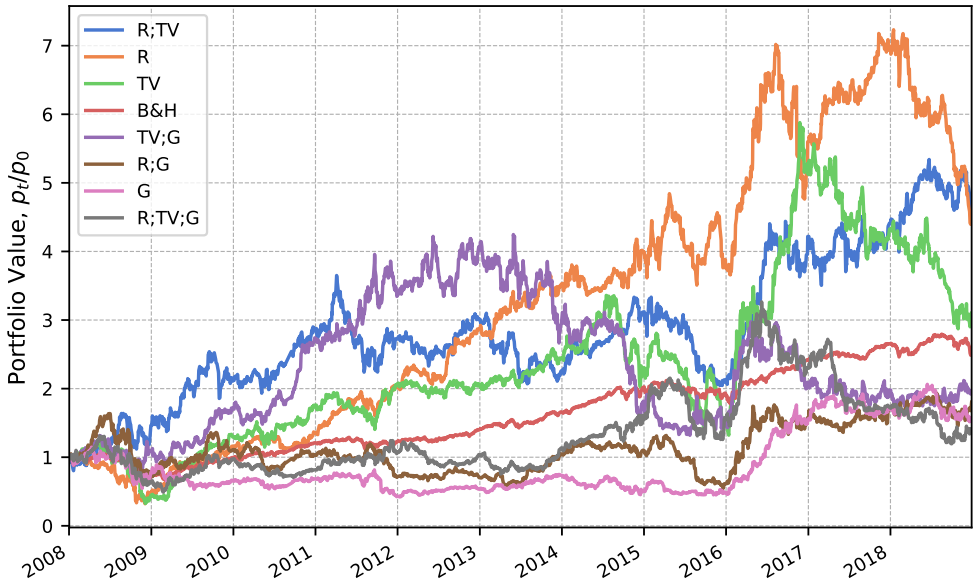


Figure A.3: Cumulative return of DDPG and buy-and-hold strategy on TSX over the time period 2008-2019. Agents trade on a daily basis and predict daily returns. $\tau = 0.00\%$. Legend sorted by descending cumulative return.

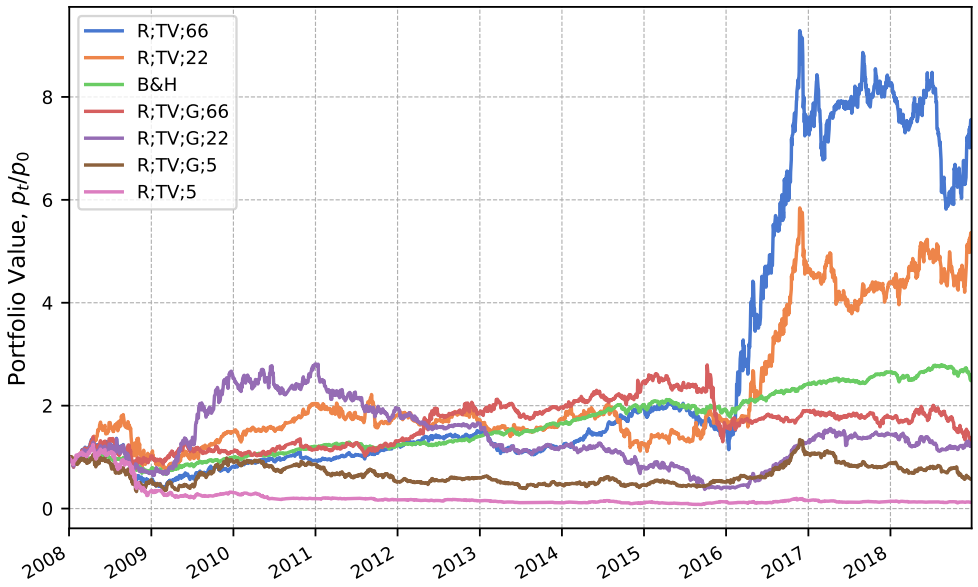


Figure A.4: Cumulative return of DDPG and buy-and-hold strategy on TSX over the time period 2008-2019. Agents are restricted to only trade once every week, month or quarter. Thus. they predict weekly, monthly or quarterly returns respectively. $\tau = 0.25\%$. Legend sorted by descending cumulative return.

Table A.5: Daily metrics for DDPG for each combination of predictors, and the buy-and-hold strategy, on TSX 2008-2019.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	0.06	0.04	0.02	0.06	0.02	0.03	0.01	0.03
Std. Dev. [%]	2.18	2.47	2.23	1.95	2.36	2.30	2.10	0.95
Skewness	-1.36	-0.12	-0.35	-0.06	0.24	0.22	-0.02	-0.28
Kurtosis	23.50	15.01	8.34	5.66	10.23	5.60	4.92	9.74
$VaR_{5\%}$ [%]	2.89	3.61	3.36	2.94	3.63	3.53	3.23	1.44
$VaR_{95\%}$ [%]	2.83	3.44	3.25	3.06	3.51	3.54	3.29	1.32
$CVaR_{5\%}$ [%]	5.42	5.87	5.48	4.69	5.57	5.23	4.87	2.33
$CVaR_{95\%}$ [%]	5.13	5.92	5.21	4.72	5.70	5.55	5.04	2.13
Hit Ratio	0.53	0.50	0.51	0.51	0.51	0.49	0.50	0.55
Avg. Returns+ [%]	1.27	1.57	1.46	1.33	1.52	1.63	1.44	0.61
Avg. Returns- [%]	-1.35	-1.56	-1.52	-1.32	-1.57	-1.57	-1.48	-0.66

Table A.6: Annualized metrics for DDPG run on different set of predictors, and buy-and-hold strategy on TSX.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	14.03	10.40	4.17	14.30	5.52	6.33	2.99	8.80
Std. Dev. [%]	34.65	39.22	35.43	30.96	37.44	36.57	33.33	15.07
Sharpe Ratio	0.41	0.27	0.12	0.46	0.15	0.17	0.09	0.58
FF-5 alpha [%]	18.37	15.42	8.55	17.66	12.96	13.64	8.42	7.97

Dataset: Fama/French North America 5 Factors [Daily]

Table A.7: Daily metrics for DDPG run with trading horizon and Buy-and-Hold strategy on TSX. Trading horizons are either 0, 5, 22, or 66 days. $\tau = 0.25\%$.

	5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	-0.08	-0.02	0.06	0.01	0.07	0.01	0.03
Std. Dev. [%]	2.59	2.46	2.41	2.32	2.34	1.79	0.94
Skewness	-0.99	-0.76	-0.47	0.21	0.42	0.08	-0.31
Kurtosis	11.23	13.39	10.41	5.39	9.64	7.47	9.78
$VaR_{5\%}$ [%]	4.04	3.82	3.46	3.64	3.50	2.82	1.44
$VaR_{95\%}$ [%]	3.86	3.62	3.72	3.70	3.50	2.59	1.31
$CVaR_{5\%}$ [%]	6.71	5.89	5.84	5.36	5.60	4.36	2.32
$CVaR_{95\%}$ [%]	5.86	5.79	5.80	5.72	5.84	4.43	2.10
Hit Ratio	0.48	0.49	0.51	0.49	0.52	0.50	0.55
Avg. Returns+ [%]	1.64	1.61	1.57	1.59	1.51	1.16	0.60
Avg. Returns- [%]	-1.71	-1.65	-1.57	-1.57	-1.50	-1.18	-0.66

Table A.8: Annualized metrics for DDPG run with trading horizon and buy-and-hold strategy on TSX. Trading horizons are either 5, 22, or 66 days. $\tau = 0.25\%$.

	5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	-19.35	-5.23	15.50	2.07	18.68	2.91	8.73
Std. Dev. [%]	41.07	39.00	38.23	36.76	37.21	28.41	14.97
Sharpe Ratio	-0.47	-0.13	0.41	0.06	0.50	0.10	0.58
FF-5 alpha [%]	-12.25	2.11	21.15	8.32	24.06*	4.78	7.72

Dataset: Fama/French North America 5 Factors [Daily]

Note: * $p < .1$ statistically different from B&H

A.3.3 JSE Securities Exchange Index

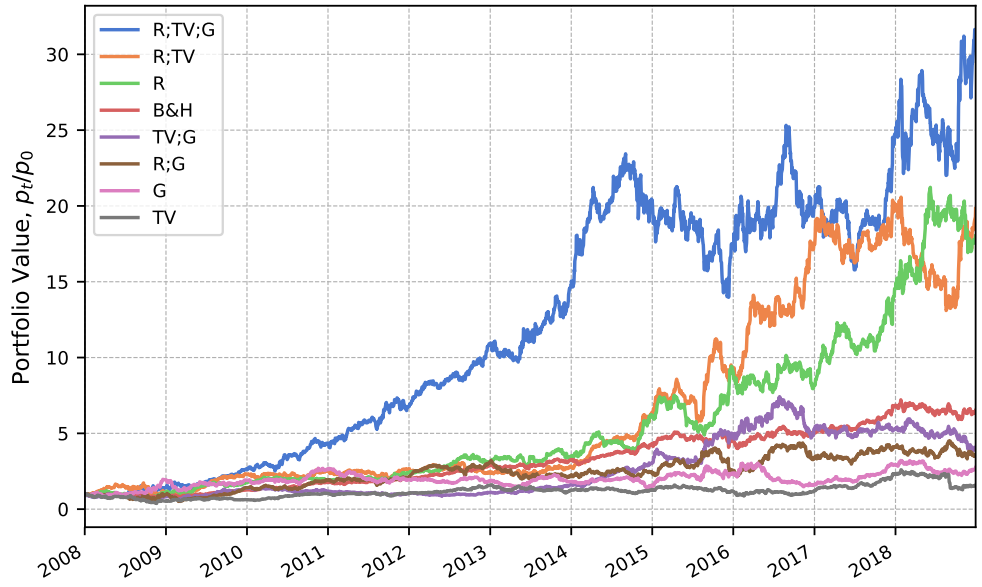


Figure A.5: Cumulative return of DDPG and buy-and-hold strategy on JSE over the time period 2008-2019. Agents trade on a daily basis and predict daily returns. $\tau = 0.0\%$. Legend sorted by descending cumulative return.

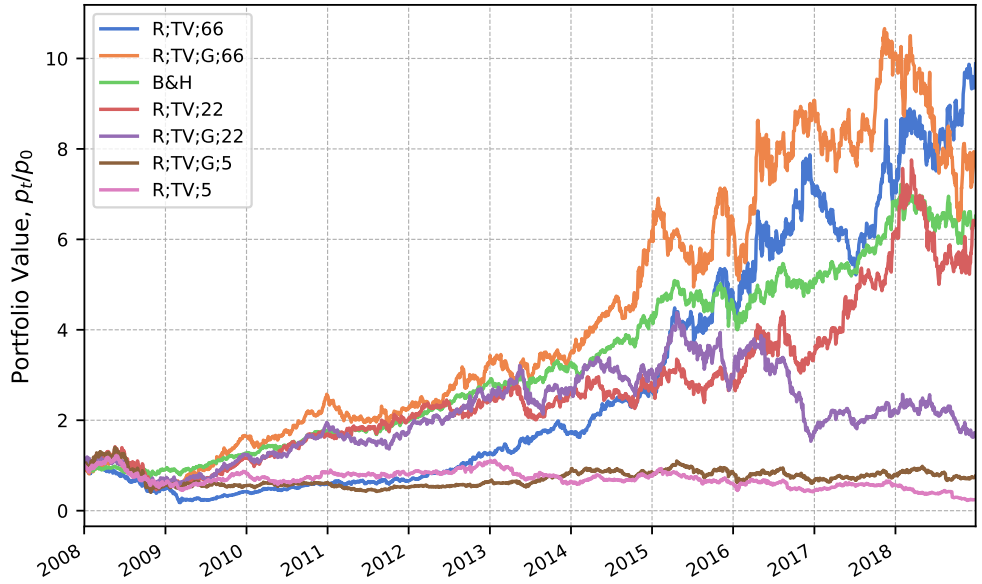


Figure A.6: Cumulative return of DDPG and buy-and-hold strategy on JSE over the time period 2008-2019. Agents are restricted to only trade once every week, month or quarter. Thus. they predict weekly, monthly or quarterly returns respectively. $\tau = 0.25\%$. Legend sorted by descending cumulative return.

Table A.9: Daily metrics for DDPG for each combination of predictors, and the buy-and-hold strategy, on JSE 2008-2019.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	0.10	0.02	0.04	0.11	0.05	0.05	0.12	0.07
Std. Dev. [%]	2.00	2.11	2.11	2.11	2.25	1.98	2.04	1.16
Skewness	0.24	-0.46	0.61	-0.27	0.08	-0.00	0.18	-0.12
Kurtosis	3.97	12.13	5.60	6.29	7.99	3.79	4.72	2.28
$VaR_{5\%}$ [%]	3.06	3.12	3.19	2.95	3.18	3.00	3.11	1.87
$VaR_{95\%}$ [%]	3.41	2.97	3.29	3.32	3.38	3.15	3.30	1.85
$CVaR_{5\%}$ [%]	4.40	4.87	4.59	4.85	5.19	4.55	4.44	2.58
$CVaR_{95\%}$ [%]	4.93	4.87	5.22	4.98	5.26	4.68	4.93	2.63
Hit Ratio	0.52	0.50	0.49	0.53	0.50	0.50	0.52	0.53
Avg. Returns+ [%]	1.42	1.44	1.53	1.49	1.55	1.43	1.47	0.88
Avg. Returns- [%]	-1.39	-1.44	-1.43	-1.47	-1.54	-1.39	-1.41	-0.88

Table A.10: Annualized metrics for DDPG run on different set of predictors, and buy-and-hold strategy on JSE.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	25.88	4.03	9.03	26.73	11.44	12.55	30.79	16.77
Std. Dev. [%]	31.79	33.53	33.46	33.53	35.74	31.45	32.36	18.48
Sharpe Ratio	0.81	0.12	0.27	0.80	0.32	0.40	0.95	0.91
FF-5 alpha [%]	27.64	6.75	12.24	29.13	16.82	12.20	32.04*	15.71

Dataset: Fama/French Global 5 Factors [Daily]

Note: * $p < .1$ statistically different from B&H

Table A.11: Daily metrics for DDPG run with trading horizon and buy-and-hold strategy on JSE. Trading horizons are either 5, 22, or 66 days. $\tau = 0.25\%$.

	5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	-0.05	-0.01	0.07	0.02	0.08	0.07	0.07
Std. Dev. [%]	2.11	2.13	2.13	2.03	2.29	2.00	1.16
Skewness	-0.53	-0.55	0.07	-0.10	-0.46	0.04	-0.12
Kurtosis	7.13	6.87	3.14	2.57	8.40	3.85	2.28
$VaR_{5\%}$ [%]	3.46	3.25	3.35	3.39	3.43	3.20	1.87
$VaR_{95\%}$ [%]	3.12	3.24	3.43	3.22	3.35	3.22	1.85
$CVaR_{5\%}$ [%]	5.02	5.17	4.88	4.80	5.51	4.66	2.58
$CVaR_{95\%}$ [%]	4.66	4.72	4.97	4.65	5.14	4.71	2.63
Hit Ratio	0.49	0.49	0.50	0.50	0.51	0.51	0.53
Avg. Returns+ [%]	1.46	1.48	1.58	1.44	1.59	1.44	0.88
Avg. Returns- [%]	-1.52	-1.48	-1.50	-1.50	-1.64	-1.44	-0.88

Table A.12: Annualized metrics for DDPG run with trading horizon and buy-and-hold strategy on JSE. Trading horizons are either 5, 22, or 66 days. $\tau = 0.25\%$.

	5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	-12.60	-2.40	16.59	4.83	20.50	18.28	16.77
Std. Dev. [%]	33.56	33.74	33.83	32.15	36.28	31.81	18.48
Sharpe Ratio	-0.38	-0.07	0.49	0.15	0.57	0.57	0.91
FF-5 alpha [%]	-9.70	1.66	18.62	7.02	24.46	19.80	15.71

Dataset: Fama/French Global 5 Factors [Daily]

A.3.4 S&P BSE SENSEX

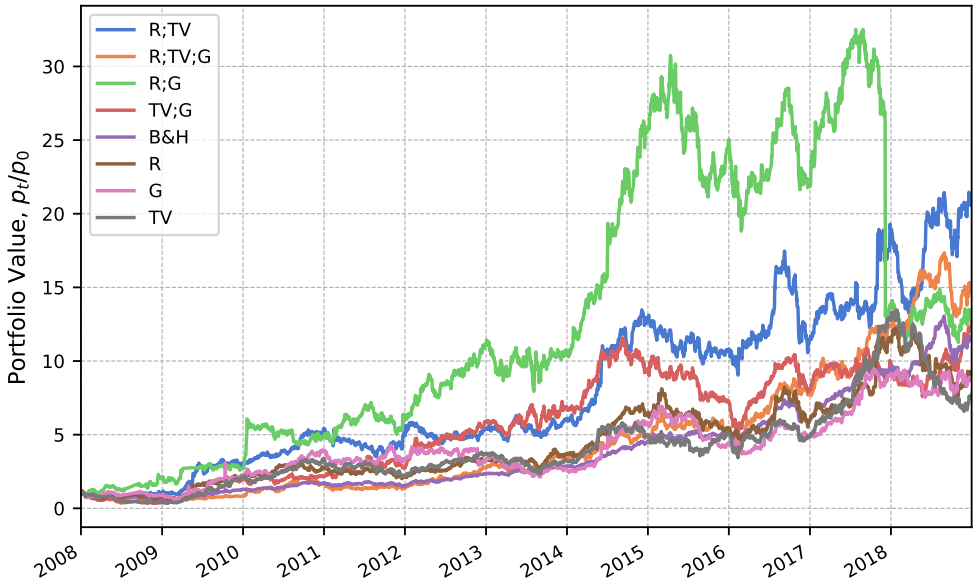


Figure A.7: Cumulative return of DDPG and buy-and-hold strategy on SENSEX over the time period 2008-2019. Agents trade on a daily basis and predict daily returns. $\tau = 0.00\%$. Legend sorted by descending cumulative return.

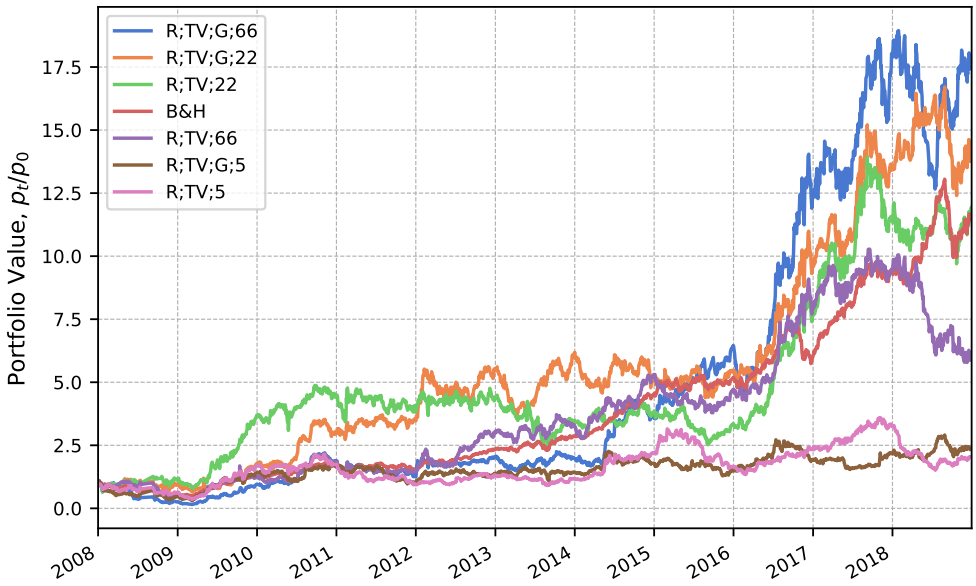


Figure A.8: Cumulative return of DDPG and buy-and-hold strategy on SENSEX over the time period 2008-2019. Agents are restricted to only trade once every week, month or quarter. Thus. they predict weekly, monthly or quarterly returns respectively. $\tau = 0.25\%$. Legend sorted by descending cumulative return.

Table A.13: Daily metrics for DDPG for each combination of predictors, and the buy-and-hold strategy, on SENSEX 2008-2019.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	0.08	0.08	0.08	0.11	0.10	0.10	0.10	0.09
Std. Dev. [%]	2.44	2.31	2.72	2.35	2.62	3.05	2.12	1.45
Skewness	4.57	0.66	0.39	1.08	-6.17	2.54	0.65	-0.15
Kurtosis	85.59	6.11	69.82	8.59	197.41	152.59	6.14	30.42
$VaR_{5\%}$ [%]	3.30	3.46	3.16	3.22	3.12	3.61	3.13	2.01
$VaR_{95\%}$ [%]	3.36	3.60	3.59	3.71	3.63	3.85	3.53	2.10
$CVaR_{5\%}$ [%]	4.69	4.97	5.36	4.76	4.86	5.67	4.60	3.22
$CVaR_{95\%}$ [%]	5.87	5.80	6.09	6.08	5.90	6.64	5.21	3.20
Hit Ratio	0.49	0.50	0.49	0.49	0.50	0.49	0.51	0.54
Avg. Returns+ [%]	1.63	1.68	1.70	1.73	1.59	1.85	1.56	0.98
Avg. Returns- [%]	-1.46	-1.56	-1.50	-1.50	-1.50	-1.66	-1.45	-0.95

Table A.14: Annualized metrics for DDPG run on different set of predictors and buy-and-hold strategy on SENSEX.

	R	TV	G	R;TV	R;G	TV;G	R;TV;G	B&H
Mean Returns [%]	21.23	19.46	20.83	28.83	24.23	24.18	25.81	23.42
Std. Dev. [%]	38.78	36.73	43.24	37.36	41.57	48.38	33.69	23.06
Sharpe Ratio	0.55	0.53	0.48	0.77	0.58	0.50	0.77	1.02
FF-5 alpha [%]	24.62	21.78	25.03	32.40	26.60	28.81	27.30	21.97

Dataset: Fama/French Asia Pasific ex Japan 5 Factors [Daily]

Table A.15: Daily metrics for DDPG run with trading horizon and buy-and-hold strategy on SENSEX. Trading horizons are either 0, 5, 22, or 66 days. $\tau = 0.25\%$.

	5		22		66		B&H
	R;TV	R;TV;G	R;TV	R;TV;G	R;TV	R;TV;G	
Mean Returns [%]	0.03	0.03	0.09	0.10	0.07	0.11	0.09
Std. Dev. [%]	2.40	2.40	2.40	2.55	2.59	2.85	1.45
Skewness	0.59	0.57	0.64	1.14	-0.82	-0.86	-0.15
Kurtosis	6.35	6.91	11.22	37.40	39.92	21.47	30.42
$VaR_{5\%}$ [%]	3.66	3.64	3.39	3.43	3.43	3.80	2.01
$VaR_{95\%}$ [%]	3.89	3.67	3.84	3.86	3.86	4.42	2.10
$CVaR_{5\%}$ [%]	5.38	5.42	4.99	5.31	5.54	6.21	3.22
$CVaR_{95\%}$ [%]	5.84	5.85	6.04	6.11	5.91	6.90	3.20
Hit Ratio	0.48	0.49	0.48	0.50	0.48	0.50	0.54
Avg. Returns+ [%]	1.76	1.69	1.78	1.74	1.77	2.00	0.98
Avg. Returns- [%]	-1.59	-1.62	-1.54	-1.60	-1.55	-1.78	-0.95

Table A.16: Annualized metrics for DDPG run with trading horizon and buy-and-hold strategy on SENSEX. Trading horizons are either 5, 22, or 66 days. $\tau = 0.25\%$.

	5		22		66		B&H
	R;V	R;V;G	R;V	R;V;G	R;V	R;V;G	
Mean Returns [%]	6.72	8.30	23.63	25.28	17.18	27.39	23.42
Std. Dev. [%]	38.18	38.04	38.16	40.46	41.05	45.17	23.06
Sharpe Ratio	0.18	0.22	0.62	0.62	0.42	0.61	1.02
FF-5 alpha [%]	8.89	8.82	25.23	28.99	22.69	30.86	21.97

Dataset: Fama/French Asia Pasific ex Japan 5 Factors [Daily]

A.4 Descriptive Statistics for Index Component Stocks

Descriptive statistics for DJIA (USA), TSX (Canada), JSE (South Africa) and SENSEX (India) are summarized in this section. Figure A.9 shows the cumulative return of a buy-and-hold portfolio on each market. The stocks considered in each market and their descriptive statistics are displayed in tables A.17, A.18, A.19 and A.20. There, the descriptive statistics of the buy-and-hold strategy can also be found.

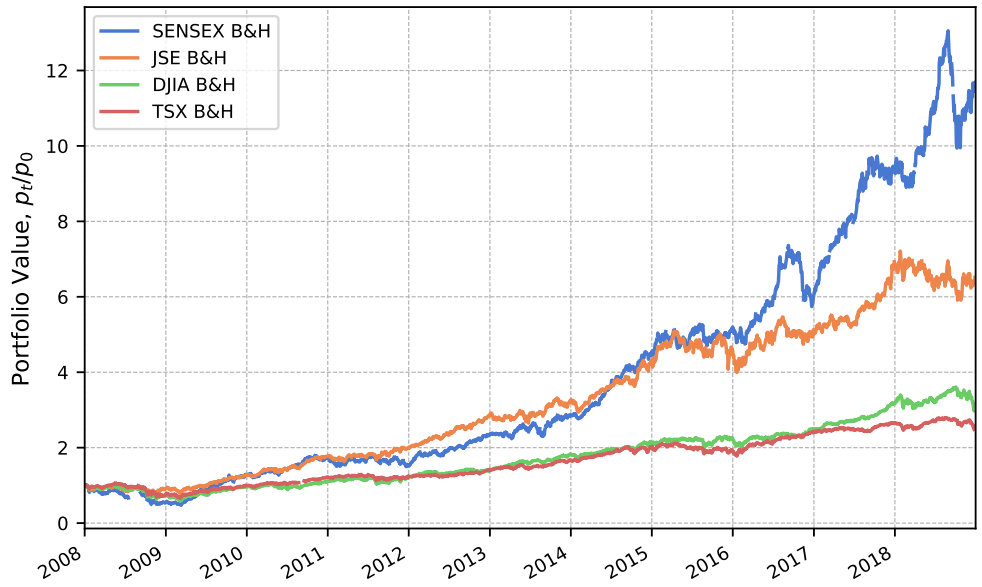


Figure A.9: Cumulative return of buy-and-hold strategy on DJIA, SENSEX, TSX and JSE. Legend sorted by descending cumulative return.

Table A.17: Descriptive statistics for each stock considered from DJIA, with buy-and-hold portfolio included. The time period is from 2004 to 2019. The Sharpe Ratio is annualized.

	mean [%]	std [%]	min [%]	max [%]	SR	VaR [%]	ES [%]	Cum. rtn [%]
MMM	0.03	1.34	-9.38	9.42	0.38	1.98	3.36	119.98
AXP	0.03	2.14	-19.35	18.77	0.21	3.01	5.29	104.79
AAPL	0.13	2.10	-19.75	13.02	1.01	3.11	4.73	504.32
BA	0.06	1.71	-9.35	14.38	0.59	2.56	4.05	239.08
CAT	0.04	1.96	-15.69	13.73	0.33	3.04	4.73	155.84
CVX	0.04	1.60	-13.34	18.94	0.39	2.45	3.73	147.81
CSCO	0.02	1.81	-17.69	14.80	0.19	2.61	4.31	80.13
KO	0.03	1.08	-9.07	13.00	0.51	1.56	2.53	131.19
DIS	0.05	1.60	-10.23	14.82	0.46	2.29	3.71	173.89
XOM	0.02	1.47	-15.03	15.86	0.26	2.17	3.42	91.64
GS	0.02	2.21	-21.02	23.48	0.14	3.10	5.21	71.60
HD	0.05	1.57	-8.58	13.16	0.52	2.41	3.60	193.74
IBM	0.01	1.33	-8.66	10.90	0.18	2.01	3.28	56.26
INTC	0.02	1.80	-13.22	11.20	0.18	2.75	4.24	78.20
JNJ	0.04	1.01	-10.58	11.54	0.56	1.47	2.34	133.79
JPM	0.04	2.35	-23.23	22.39	0.25	3.09	5.50	137.51
MCD	0.06	1.20	-8.32	8.97	0.84	1.82	2.64	240.36
MRK	0.03	1.64	-31.17	12.25	0.27	2.19	3.84	106.09
MSFT	0.04	1.63	-12.46	17.06	0.42	2.35	3.81	162.02
NKE	0.08	1.66	-12.60	11.88	0.76	2.41	3.70	299.06
PFE	0.02	1.39	-11.82	9.69	0.24	2.02	3.19	78.50
PG	0.03	1.06	-8.23	9.73	0.42	1.55	2.52	105.03
TRV	0.04	1.67	-20.07	22.76	0.37	2.20	3.88	147.98
UTX	0.03	1.41	-9.17	12.79	0.35	2.12	3.32	116.42
UNH	0.06	1.94	-20.62	29.83	0.49	2.66	4.40	227.59
VZ	0.04	1.28	-8.41	13.66	0.43	1.90	2.92	132.21
WMT	0.02	1.21	-10.74	10.50	0.31	1.77	2.77	89.93
WBA	0.02	1.63	-16.24	15.39	0.23	2.36	3.76	89.90
DJIA B&H	0.05	1.07	-8.40	11.60	0.69	1.60	2.58	167.06

Table A.18: Descriptive statistics for each stock considered from TSX, with buy-and-hold portfolio included. The time period is from 2004 to 2019. The Sharpe Ratio is annualized.

	mean [%]	std [%]	min [%]	max [%]	SR	VaR [%]	ES [%]	Cum. rtn [%]
TD.TO	0.05	1.29	-13.63	12.34	0.66	1.77	2.96	199.75
BNS.TO	0.04	1.31	-14.31	12.05	0.43	1.91	3.11	131.59
ENB.TO	0.05	1.31	-10.56	9.81	0.66	1.99	2.87	204.81
CNR.TO	0.07	1.44	-11.78	10.02	0.72	2.16	3.26	245.24
BMO.TO	0.03	1.31	-13.08	11.79	0.38	1.88	3.19	117.72
CM.TO	0.03	1.40	-13.29	13.15	0.35	1.98	3.37	115.62
CNQ.TO	0.04	2.44	-24.22	16.08	0.29	3.66	5.52	166.78
MFC.TO	0.01	2.00	-16.63	16.77	0.10	2.82	4.94	46.39
BCE.TO	0.04	1.33	-41.79	12.02	0.42	1.49	2.73	133.28
BAM-A.TO	0.06	1.75	-22.65	24.18	0.58	2.48	4.00	237.24
TRP.TO	0.03	1.16	-9.13	8.94	0.43	1.78	2.69	117.72
CP.TO	0.06	1.76	-12.08	14.62	0.50	2.63	4.00	208.54
IMO.TO	0.02	1.78	-15.23	15.93	0.19	2.61	4.12	80.20
SLF.TO	0.02	1.80	-15.04	17.88	0.22	2.38	4.33	92.62
RCI-B.TO	0.06	1.46	-10.48	10.46	0.67	2.08	3.33	229.72
T.TO	0.06	1.35	-14.53	12.92	0.67	1.93	2.98	213.19
L.TO	0.01	1.22	-13.26	12.83	0.16	1.73	2.74	47.58
WCN.TO	0.06	1.69	-21.61	10.01	0.52	2.18	3.92	207.85
NA.TO	0.05	1.37	-18.23	13.65	0.62	1.85	3.13	199.01
PPL.TO	0.06	1.50	-17.77	19.40	0.61	2.16	3.45	214.67
TECK-B.TO	0.03	3.48	-27.74	31.40	0.16	5.06	8.07	129.04
ABX.TO	-0.01	2.66	-17.91	28.11	-0.05	3.95	6.19	-29.72
GIB-A.TO	0.06	1.69	-8.14	16.57	0.59	2.46	3.56	234.17
HSE.TO	0.02	1.88	-13.73	12.40	0.15	2.99	4.35	68.33
FTS.TO	0.05	1.20	-10.81	9.54	0.66	1.78	2.70	187.46
SAP.TO	0.06	1.43	-9.07	12.80	0.65	2.07	3.18	220.28
ECA.TO	-0.01	2.52	-15.49	20.42	-0.04	4.00	5.92	-25.79
POW.TO	0.02	1.51	-14.72	14.23	0.17	2.22	3.50	60.89
WN.TO	0.00	1.26	-12.48	8.47	0.05	1.89	2.88	15.05
AEM.TO	0.04	2.92	-28.41	19.22	0.20	4.31	6.59	136.98
TSX B&H	0.04	0.92	-7.99	6.74	0.65	1.43	2.26	132.67

Table A.19: Descriptive statistics for each stock considered from JSE, with buy-and-hold portfolio included. The time period is from 2004 to 2019. The Sharpe Ratio is annualized.

	mean [%]	std [%]	min [%]	max [%]	SR	VaR [%]	ES [%]	Cum. rtn [%]
NPN.JO	0.11	2.06	-10.78	10.38	0.86	3.36	4.54	432.43
BHP.JO	0.05	2.18	-11.42	18.00	0.38	3.40	4.90	201.70
CFR.JO	0.05	1.97	-59.36	9.45	0.40	2.61	4.13	192.96
AMS.JO	0.02	2.68	-17.59	17.92	0.14	4.15	6.14	90.23
SOL.JO	0.05	2.05	-11.45	11.43	0.41	3.17	4.69	202.39
SBK.JO	0.06	1.83	-14.55	10.42	0.48	2.89	4.06	212.91
FSR.JO	0.08	1.95	-16.06	12.24	0.63	3.04	4.27	298.89
MTN.JO	0.04	2.23	-21.59	16.25	0.32	3.28	4.90	170.48
SLM.JO	0.07	1.79	-11.27	11.87	0.64	2.87	4.04	279.36
ABG.JO	0.05	1.85	-15.70	11.22	0.45	2.82	4.14	203.03
REM.JO	0.07	1.54	-9.20	10.13	0.75	2.39	3.23	280.98
SHP.JO	0.09	1.77	-9.21	11.54	0.79	2.73	3.80	338.91
APN.JO	0.07	1.93	-19.56	10.20	0.54	2.87	4.35	254.58
BVT.JO	0.11	1.74	-10.05	16.69	1.00	2.66	3.75	422.10
MRP.JO	0.10	1.95	-19.63	9.50	0.85	2.89	4.33	404.51
NED.JO	0.05	1.81	-11.15	11.84	0.47	2.79	3.95	207.98
TBS.JO	0.05	1.61	-20.85	8.03	0.53	2.47	3.64	205.10
WHL.JO	0.07	1.79	-10.29	12.84	0.64	2.88	3.92	276.92
SAP.JO	0.02	2.33	-22.31	16.31	0.11	3.37	5.37	63.15
RMH.JO	0.08	1.99	-13.32	11.47	0.61	3.13	4.34	295.19
DSY.JO	0.08	1.67	-15.40	9.85	0.71	2.60	3.69	289.25
CLS.JO	0.09	1.72	-8.41	9.33	0.85	2.71	3.70	355.50
CPI.JO	0.15	1.91	-12.26	13.49	1.25	2.86	4.37	578.10
ANG.JO	-0.01	2.59	-15.82	17.56	-0.06	4.05	5.68	-35.83
NTC.JO	0.06	1.71	-10.59	9.24	0.51	2.63	3.83	212.39
GFI.JO	-0.01	2.85	-16.03	19.39	-0.04	4.33	6.43	-29.61
TFG.JO	0.07	1.95	-11.76	13.03	0.60	3.04	4.22	283.92
TRU.JO	0.08	2.04	-12.46	11.10	0.59	3.15	4.41	291.05
JSE B&H	0.07	1.06	-6.29	5.48	1.11	1.67	2.41	278.53

Table A.20: Descriptive statistics for each stock considered from SENSEX, with buy-and-hold portfolio included. The time period is from 2004 to 2019. The Sharpe Ratio is annualized.

	mean [%]	std [%]	min [%]	max [%]	SR	VaR [%]	ES [%]	Cum. rtn [%]
ASIANPAINT.BO	0.15	1.89	-13.57	25.34	1.27	2.45	3.77	546.87
AXISBANK.BO	0.11	2.66	-17.88	20.06	0.64	3.96	5.78	389.02
BAJFINANCE.BO	0.24	3.56	-22.05	127.29	1.06	3.62	5.79	863.29
BHARTIARTL.BO	0.05	2.30	-14.14	15.26	0.34	3.48	5.12	178.72
HDFCBANK.BO	0.10	1.95	-23.10	35.84	0.83	2.63	4.03	370.43
HCLTECH.BO	0.09	2.35	-16.36	17.15	0.62	3.41	5.40	332.88
HEROMOTOCO.BO	0.07	1.96	-9.22	27.82	0.53	2.87	4.15	238.38
HINDCOPPER.BO	0.01	3.46	-25.84	26.43	0.04	5.08	6.87	28.96
HDFC.BO	0.09	2.24	-11.73	32.35	0.67	3.12	4.75	342.95
ICICIBANK.BO	0.08	2.61	-21.96	31.62	0.50	3.74	5.61	301.42
INDUSINDBK.BO	0.10	2.92	-19.99	24.87	0.56	4.13	6.50	377.23
ITC.BO	0.18	4.85	-40.37	209.86	0.60	2.71	4.36	667.91
KOTAKBANK.BO	0.12	2.60	-23.22	32.86	0.74	3.67	5.81	437.60
LT.BO	0.09	2.67	-38.60	40.55	0.52	3.46	5.43	316.78
M&M.BO	0.10	3.54	-68.52	69.32	0.45	3.43	6.07	367.55
MARUTI.BO	0.08	2.15	-20.38	26.42	0.63	3.10	4.73	308.74
ONGC.BO	0.07	2.64	-39.46	40.55	0.43	3.18	5.13	257.78
RELIANCE.BO	0.08	2.09	-17.96	19.53	0.63	3.07	4.71	300.30
SBIN.BO	0.10	2.57	-18.27	31.53	0.64	3.54	5.14	379.69
SUNPHARMA.BO	0.09	2.03	-16.19	23.37	0.73	2.92	4.43	340.37
TCS.BO	0.09	2.63	-69.79	69.17	0.56	2.83	5.03	336.22
TATAMOTORS.BO	0.04	2.69	-18.08	17.49	0.26	3.99	6.06	160.48
TATASTEEL.BO	0.03	2.72	-16.24	14.96	0.19	4.19	6.44	117.99
VEDL.BO	0.21	5.13	-25.08	220.44	0.66	4.65	6.89	779.04
SENSEX B&H	0.11	1.47	-10.69	26.03	1.16	2.02	3.22	365.40