ROBUST FOREX TRADING WITH DEEP Q NETWORK (DQN)

Sutta Sornmayura*

Abstract

Financial trading is one of the most attractive areas in finance. Trading systems development is not an easy task because it requires extensive knowledge in several areas such as quantitative analysis, financial skills, and computer programming. A trading systems expert, as a human, also brings in their own bias when developing the system. There should be another, more effective way to develop the system using artificial intelligence. The aim of this study was to compare the performance of AI agents to the performance of the buy-and-hold strategy and the expert trader. The tested market consisted of 15 years of the Forex data market, from two currency pairs (EURUSD, USDJPY) obtained from Dukascopy Bank SA Switzerland. Both hypotheses were tested with a paired t-Test at the 0.05 significance level. The findings showed that AI can beat the buy & hold strategy with significant superiority, in FOREX for both currency pairs (EURUSD, USDJPY), and that AI can also significantly outperform CTA (experienced trader) for trading in EURUSD. However, the AI could not significantly outperform CTA for USDJPY trading. Limitations, contributions, and further research were recommended.

Keywords: Financial Trading, Artificial Intelligence, Machine Learning, Forex Market, Gold Market, Deep Q Network, Deep Q Learning, Deep Learning

*Sutta Sornmayura obtains a B.Eng. in Electrical Engineering from King Mongkut's University of Technology Thonburi (KMUTT), a M.Eng.Sc from The University of New South Wales (Biomedical Engineering), and MBA from Thammasat University. He is a Ph.D. candidate in International College, National Institute of Development Administration(NIDA). Currently, he is working as a full-time lecturer in the School of Management, Department of Industrial Management and Logistics at Assumption University. He is also a financial trading system developer for leading stock brokerage in Thailand with extensive expertise in Artificial Intelligence, Machine Learning, Data Science, and Blockchain Technology.

NOMENCLATURE

AI	Artificial Intelligence
DQN	Deep Q Network
CTA	Commodity Trading Advisor
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
EMH	Efficient Market Hypothesis

1. INTRODUCTION

Financial trading is one of the most interesting, challenging, and promising areas in finance. Developing a profitable trading system is a difficult task for professional traders, but there is a high reward pay-off for such a dedicated endeavor. Currently, there are two schools of thought when conducting trading systems development, based on the value in the long term fundamental factors called fundamental analysis or based on future price movement that possesses predictive power called technical analysis (Lo, Mamaysky, & Wang, 2000).

Several kinds of literature substantiate the idea that technical analysis, if appropriately applied, can be used to develop a profitable system and which possesses a statistical edge to trade in the market. One study of technical analysis and the fuzzy logic application, using three technical indicators, ROC, Stochastic, and support/resistance to study four stocks, found the application to be excellent, surpassing S&P500 performance (Dourra & Siy, 2002). Another study looked at the application of a neural network using the technical indicators, SMA, stochastics, and momentum. In this study, Chan et al. (1995) studied the neural network in predicting the trading signal before the crowd joined the trade. The result was found to be more profitable than relying on traditional technical signals. The study showed that if traders could predict the trading signals before the majority of traders found it, they would be able to make more money (Chan & Teong, 1995). Based on several kinds of literature, there is some predictive power in the technical analysis that uses price and technical indicators as the key to unlocking the future price and profitable opportunities. Therefore, the purpose of this study is to use the power of price and technical indicators as valuable pieces of information to uncover the hidden profitable pattern and to develop the trading system.

The market is continually changing and evolving due to non-linear relationships, chaos, and the stochastic nature of the market (Hsieh, 1991). Developing a robust trading system requires key features, which have adaptive capabilities and synchronicity with the market. Therefore, the concepts of machine learning and artificial intelligence, in which the computer could learn to automate trade from the data is very challenging to study (Kalmus, Trojan, Mott, & Strampfer, 1987).

The foreign exchange market (Forex) is a decentralized global market for the trading of currencies. The market includes all aspects of buying, selling and exchanging currencies at current or determined prices. Regarding the volume of trading, the forex market is by far the largest market in the world, followed by the credit market. The main participants in this market are large international banks and financial institutions. In this paper, the researcher found advantages to choosing the forex market over the stock market due to the following reasons: no corporate action (data cleaning is convenient) , it is a 24-hour market, it is the most significant financial market in the world, there is prevalent data excess, and it is the most liquid market in the world (Yao & Tan, 2000).

1.1 Problem Statement

There are several drawbacks when one develops a trading system, such as a bias toward certainly preferred instruments, bias to choose favorite indicators, bias to choose the in- sample period, and bias toward optimization. All of these biases are most commonly brought to the development process by the trading system developer (Merold, Malkin, Riordan, & Howorka, 2002).

There are also countless indicators and parameters to fine tune to fit the model to historical data and become profitable using a backtest process. The system developers spend most of their time with the optimization process to find the parameters which guarantee success in the live market. The system traders can end up with a trading system that allows them to trade profitably under specific market conditions and to be confident to undertake live trade. However, the trading system will fail when the market and trading system are not in synchronicity (Huang, Hung, & Yen, 2005).

There is a link between the problem of the practitioner (trading system developers) and the current gap in academic research, such that it is believed that there is no previous paper, which has successfully studied the application of computer learning to trade in the forex market. Moreover, if a computer could outperform both humans and the buy and hold strategy, this study will provide a contribution to practitioners for finding new methods in forex trading.

1.2 The objective of The Research

The purpose of this paper is to explore the possibility of creating an automated, robust trading system, using the combined knowledge from machine learning, quantitative finance, and big data computing power. We try to answer two research questions:

1. Can we teach a computer to develop a trading system that beats the buy-and-hold strategy?

2. Can a machine trader outperform an experienced trader?

The Efficient Market Hypothesis (EMH) states that it is impossible to beat the market by timing the market consistently. Therefore, the best strategy for an EMH supporter is buy-and-hold. However, if the machine can see a repeatable, profitable pattern in the market that humans cannot see, it is possible that the machine could detect hidden patterns in the market correctly or at least more quickly, allowing it to act before human traders. Finally, it would be possible to make consistent profits and provide performance which is better than the buy-and-hold strategy. We can also use the benchmark of currency index fund from BarclayHedge (BarclayHedge, 2017) to answer the question of whether the machine is better than a human expert.

The paper consists of 6 sections, Section 2 and 3 contain the literature review and research methodology, respectively. Empirical results and data analysis are reported in Section 4. The discussion and research findings are in Section 5. Lastly, Section 6 contains the conclusion, limitations, and future research possibilities.

2. LITERATURE REVIEW

2.1 Random Walk Theory

The Random Walk model believes that successive price changes are independent of each other, such that it is impossible to consistently make profits from the market by using technical analysis and fundamental analysis (Fama, 1995). The model explains that stock price is purely random and unpredictable; however, this paper will contradict this model by showing that AI can learn the hidden patterns in historical data and make a profitable decision based on these patterns.

2.2 Efficient Market Hypothesis

The efficient market hypothesis (EMH) states that the price of the security fairly and fully reflects all available information. A direct implication of EMH is that accurate timing of buying and selling in the market is purely random due to the random walk of the stock price, and there is no one who can earn a consistent abnormal return from trading.

Fama laid the foundation regarding the efficient market hypothesis, stating that all investors can easily access the same public information, so finally, nobody will be able to earn abnormal returns consistently. Profitable trades, from time to time, could be possibly a fluke. According to EMH, The investors will react to the market instantly so the profit opportunity will disappear (Malkiel & Fama, 1970). Proponents of the EMH, therefore, suggest that the most appropriate strategy to trade the market is to buy-and-hold.

The researcher found that there are several studies supporting the idea that active trading can out-perform the buy-and-hold strategy, those studies have also been shown to support the idea that technical analysis can be utilized to develop profitable trading systems. Taylor et al. (1992) conducted a questionnaire survey on professional foreign exchange dealers in Hong Kong and found that technical analysis is more useful when used in the short-term, with most of the respondents using it for forecasting the trend and turning points (Taylor & Allen, 1992). Neely et al. (1997) used a genetic programming technique to find technical trading rules and found substantial evidence to support out-of- sample excess returns for six exchange rates from 1981- 1995 (Neely, Weller, & Dittmar, 1997). Moreover, Lu et al. (2012) investigated the application of the candlestick reversal pattern which is the relationship of the open, high, low, and the closing price of stock in Taiwan during 2002-2008. All three bullish reversal patterns were profitable when applied to the stock market. (Lu, Shiu, & Liu, 2012)

Moreover, there is a study providing empirical evidence from the FOREX market, which contradicts the efficient market hypothesis. Alonso et al. (2015) have conducted a study of automated trading in the forex market. The study was conducted for six currency pairs which were EUR/USD, GBP/USD, USD/CAD, USD/JPY, USD/CHF, AUD/USD, with an optimized period from 2001- 2008, and the testing period from 2008-2011, the indicator used for generating the signal was MACD. The study showed satisfactory results for all currencies, results from all currencies showed positive returns, whereas ETF showed negative returns in some years. This study contradicts the efficient market hypothesis (Alonso-González, Peris-Ortiz, & Almenar-Llongo, 2015). The paper showed that it is possible to earn abnormal returns in the forex market by using a technical analysis developed from historical price data.

2.3 Artificial Intelligence (AI)

AI is the ability of digital computers or computer-controlled robots to solve a problem which is typically associated with the higher intellectual processing capability of a human. (Ertel, 2018)

2.4 Commodity Trading Advisor (CTA)

Nasdaq provides the definition of CTA as “An investment manager that focuses on long and short trading in the future markets. The trades are often intraday trades. Sometimes referred to as Managed Futures” (Nasdaq, 2018).

2.5 Reinforcement Learning Concept and Terminology

Reinforcement learning is one of the approaches in machine learning and states that a machine can learn a sequential decision-making process from data. There are states (features that the agent can sense from the environment), actions, and rewards composed from the environment. The agent will learn to find the optimal policy (what action to take in each specific state) that maximizes the cumulative future reward. Agents sometimes are called learners or decision makers (Whiteson, 2010).

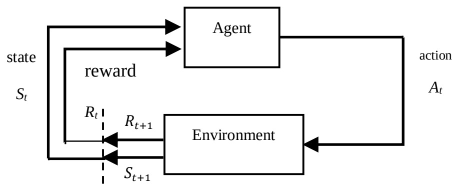

Figure 1. Reinforcement Learning Framework

(St = state at time *t*, Rt = reward at time *t*, A_t = action at time *t*)

State: $s_t \in S$ where S_t includes all possible states

Action: $a_t \in A(S_t)$ where $A(S_t)$ includes all actions in each state t

Final/Terminal States: The states that have no available actions are final/terminal states.

Episode: An episode is a complete play from one of the initial states to a final state.

For example, the agent randomly starts in one state (s), then chooses an action (a) to earn an immediate reward (r) and ends up at the next state (s′), where the process keeps repeating as a Markov decision process (MDP) until the agent finds the optimal policy.

Policy: A Policy is the agent’s strategy/behavior to choose an action in each state.

Optimal Policy: The optimal policy is the policy that theoretically maximizes the expectation of cumulative reward. From the definition of expectation and the law of large numbers, this policy has the highest average cumulative rewards given sufficient episodes. The objective of reinforcement learning is to train an agent such that his policy converges to the theoretical optimal policy.

2.6 Deep Q Learning

Deep Q learning belongs to the family of reinforcement learning and is a combination of 2 concepts, Q-Learning, and Deep learning.

It is known that deep-learning networks are good at learning hierarchical patterns of data, and also good at the representation of noisy data, invariant, and data with disturbance. Thus, we can use Deep Q-Learning as an approximation function to find Q(s, a)

Figure 2 shows that we can feed the input to the network (state) and calculate the predicted Q using the deep neural network. The predicted Q will be compared to the target for each specific action (in this example, there are four actions so we can have four Q values)

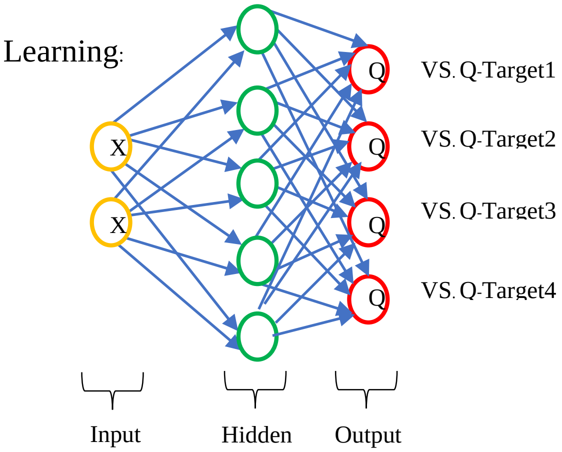

Figure 2.3 Deep Neural Network for Q learning

The loss function will be calculated as:

$$L = \frac{1}{2}[\underbrace{r + max_{a'}Q(s',a')}_{\text{Target}} - \underbrace{Q(s,a)}_{\text{Prediction}}]^2$$

Where

L is the loss function

r is reward

Q(s, a) is the Q function of state s and action a

Q(s′, a′) is the Q function of state s′ and action a′

Given a transition < *s, a, r, s'* >, the Q-table updates the rule for Q-learning in the previous algorithm and must be modified when applying the deep neural network with the following process:

1. Do a feed-forward pass for the current state, *s*, to get predicted Q-values for all actions.
2. Do a feed-forward pass for the next state, *s′*, and calculate the maximum overall network outputs $max_{a'} Q(s', a')$.
3. Set the Q-value as a target for action to $r + \gamma max_{a'} Q(s',a')$ (use the max Q-values calculated in step 2). For all other actions, set the Q-value target to be the same as initially returned from step 1, making the error 0 for those outputs.
4. Update the weights using backpropagation.

2.7 Reinforcement Learning in Financial Trading

Moody et al. (1998) studied the application of RRL (Recurrent Reinforcement Learning) in 3 empirical studies: Trader simulation, Portfolio management formulation, and an S&P500 T-bill asset allocation system. For trader simulation, they tested 2 RRL in one simulation of stock price (One for maximizing profit, one for maximizing the differential Sharpe ratio compared to the forecast model) to determine which RRL performed better. For portfolio management formulation, the RRL trained to maximize the differential Sharpe ratio performed better than that for maximizing profits. For the S&P500 T-bill asset allocation system, it showed predictive power from 1970 to 1994. (Moody, Wu, Liao, & Saffell, 1998)

Moody et al. (2001) introduced direct reinforcement learning, using the differential Sharpe ratio as a performance function for optimization. They found that direct reinforcement learning performs better than Q learning for the asset allocation problem in the S&P500 T-bill portfolio. (Moody & Saffell, 2001)

Gold (2003) studied RRL to explore the effect of training parameters on the performance of FX trading.

Dempster et al. (2006) also performed a study dealing with a usable, fully automated intelligent system. The system was based on three layers, a machine learning algorithm, risk management layer, and a dynamic optimization layer; these were collectively called Adaptive Reinforcement Learning and with the system being based on RRL. The added features made the model more flexible for different risk tolerance levels. It showed absolute profits in pips (5104) or approximately 26% p.a., compared to buy-and-hold (8% loss or 1636 pips loss) (Dempster & Leemans, 2006)

Du et al. (2016) studied the reinforcement learning method of RRL and Q learning in asset allocation problems for risky and riskless assets. The study used simulation and showed that RRL outperforms Q learning regarding stability when exposed to a noisy dataset. Q-learning is sensitive to the selection of value function. On the other hand, RRL has more flexibility to choose an objective function. (Du, Zhai, & Lv, 2016)

Deng et al. (2017) studied the performance of trading in different methods, which were FDDR, DDR, SCOT, DRL, and BH. The study used three instruments (IF, AG, SU) and used target profits (TP) and Sharpe ratio (SR) as performance functions. It found that FDDR showed the most attractive results (Deng, Bao, Kong, Ren, & Dai, 2017)

Wang et al. (2016) researched the development of an algorithmic trading system based on DQN which could automatically determine the signal to buy, sell, or hold in each trading time. (Wang et al., 2016)

After rigorous study, the researcher found that there are still no studies of the use of Deep Q network applications in the Forex market. Following the success of the Alpha Go, Deep Q network (combining deep learning with reinforcement learning) which has been applied in several areas including finance, we believe that to the best of the researchers' knowledge, this paper will be the first to explore this lucrative and most liquid market in the world.

2.8 Cumulative Annual Returns

$$\mathrm{CAGR} = \left(\frac{\text{Ending Value}}{\text{Beginning Value}}\right)^{\left(\frac{1}{\text{no. of years}}\right)} - 1$$

2.9 **Research Hypotheses**

3 AI trading performance is significantly superior to buy-and-hold performance.

4 AI trading performance is significantly superior to experienced trader performance (CTA).

3. METHODOLOGY

After rigorously reviewing several kinds of literature, the researcher found that the appropriate method to study how machines learn to trade is to translate (map) financial trading problems to a reinforcement learning problem and then train the computer through a Deep Q learning algorithm.

3.1 Mapping Reinforcement Learning (Deep Q Learning) To Financial Trading

To solve the trading problems, we need to start mapping trading problems into reinforcement problems. In order to do this, the following components need to be identified:

1. **Set of States:**

The set of states can be OHLC, indicators, and other features of the 2 instruments (EURUSD, USDJPY). This set of states represents the perceptions that the AI agent will be able to perceive in the world.

2. **Set of Actions:**

The set of actions are all the possible actions which can be taken in each state. In this case, there are 4 actions: {Hold, Buy, Sell, Close}. The agent will open only one position at a time. At any given state, the agent will choose one action.

3. **Reward Function/Performance Function:**

The reward function is the reward that the agent will receive after acting in each state. The reward function can be the function of cumulative profits (in pips), Sharpe ratio, total profits, reward to risk, etc. In this study, we will use the profits as the reward function, such as if the agent buys and the price goes up, the profit will be positive.

4. **Experience Tuple**

Experience tuple is the experience of the agent stored in the memory buffer. It is the experience of the agent that learns from the data which is < S, A, R, S′>. This part will be used for experience replay.

Using all four of the above, it is possible to find the optimal policy, π, by using a Deep- Q Learning algorithm. Training by using the Deep Q-Learning Algorithm from Mnih et al., (2013) was carried out as shown below:-

```
Initialize replay memory D to size N
Initialize action-value function Q with random weights
for episode = 1, M do
   Initialize state s_1
   for t = 1, T do
   With probability ϵ select random action a_t
   otherwise select a_t=argmax_a Q(s_t,a; θ_i)
   Execute action a_t in emulator and observe
            r_t and s_(t+1)
   Store transition (s_t, a_t, r_t, s_(t+1)) in D
   Sample a minibatch of transitions (s_j, a_j,
            r_j, s_(j+1)) from D
   Set y_j: = r_j
   for terminal s_(j+1)
   r_j+γ*max (a^' ) Q(s_(j+1),a'; θ_i) for non-
            terminal s_(j+1)
   Perform a gradient step on (y_j-Q (s_j,a_j;
          θ_i))^2 with respect to θ
   end for
end for
```

3.2 Data

The study used 15-year historical data obtained from the prominent Swiss broker, Dukascopy Bank, Switzerland. The data in our experiment is taken from the period 01/01/2001 to 12/31/2015.

The data was split into a training data set (01/01/2001-12/31/2003) and a test data set (01/01/2004 - 12/31/2015). Data was downloaded from the Dukascopy website (Dukascopy, 2017)

We used tick data obtained from a free historical feed data source converted to daily data (Dukascopy, 2017). The data were standardized for past time series change. The data were cleaned to ensure the reliability of the data. There were two currency pairs used as the universe for trade (EURUSD, USDJPY).

3.3 Experiment of Mapping Trading with Real Historical Data

We performed two experiments with real historical data (2 experiments with two currencies), with total historical data for all experiments taken from January 1, 2001, to December 31, 2015 (total 15 years). We will split the data into two sets (train/test), with the training set taken from 01/01/2001 to 12/31/2003 and the test set from 01/01/2004 to 12/31/2015. We use the following symbols to represent each currency:

- EURUSD = Euro/Dollar
- USDJPY = Dollar/Yen

The assumptions of backtesting

- The initial capital of 100,000 USD
- No transaction cost
- The position sizing is 1% for each trade
- One position can be opened at a time
- We enter using the close price of that day

Firstly, we need to map the trading problem for use as a reinforcement learning problem. Therefore, we need to specify the state, reward, and action to create the experience tuple for the agent to learn from (<S, A, R, S´>).

The performance of the AI agent also depends on what the agent perceives in its environment, which are the states that the agent can see. Typically, deep learning is good at feature extraction; it can usually detect the relevant features for classification and regression problems. However, when we set up the states which represent the features that the agent will learn, we still need human knowledge and experience to choose what to feed into the deep neural network.

States: are composed of 7 inputs

1. Close
2. Diff Close
3. Close- Sma(10) – moving average period 10
4. Close- Sma(50) – moving average period 50
5. Close- Sma(100) - moving average period 100
6. Sma(10)-Sma(50)

7. Cyclic indicators – lead sine-sine, we add sine wave indicators due to the test on a simulated sine wave which shows a positive result, so if we transform the price into the cyclic indicator like a sine wave, we believe that we would increase the performance of the AI agent dramatically.

Actions: There are 4 actions which are Buy, Sell, Close and Do Nothing.

Reward: there are intermediate rewards and long-term rewards. If the agent buys and the price goes up, the reward is the price difference. If the agent falls short and the price goes down, profits are still indicated by the price difference.

Model Configuration:

We will use a deep learning network called the ' Convolutional Neural Network -CNN' , which is widely used for image classification. For this paper, we will convert some features into a data array, feeding the data into the model.

Python Library used:

- Keras and Tensorflow (to build our neural network)
- Pyfolio from Quantopian (to create performance tear sheet)
- Jupyter notebook environment (to run the python code)

Our Brain Structure (Network topologies)

1 input layer with 7 nodes
2 hidden layers with 48 nodes
1 output layer with 4 nodes

Our activation function is 'Linear' to the output Q value

The architecture of our brain

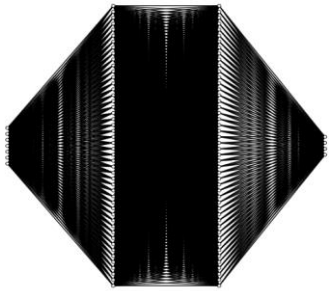

Figure 3. Architecture of Deep Neural Network (fully connected) with 7 input, 2 hidden layers (every 48 nodes), and 4 output nodes

3.4 Hypothesis Testing

There are two hypotheses for testing, which will be used to conduct the research and answer the research questions we mentioned in chapter 1.

Hypothesis 1

H0: The AI agent's performance is not superior to buy-and-hold performance.

H1: The AI agent's performance is superior to buy-and-hold performance.

Hypothesis 2

H0: The AI agent's performance is not superior to the experienced trader's performance (CTA).

H1: The AI agent's performance is superior to the experienced trader's performance (CTA).

3.5 Experimental Process

1. Data prepossessing: in this step we need to do data cleaning, preparing the database for all currency pairs.
2. Feature engineering: we need to create all relevant features which have predictive power for price movement, including all relevant indicators.
3. Split data into a training set and test set.
4. Feed data to the Deep Q-Network for learning.
5. Parameter tuning.
6. Hypothesis testing.
7. Evaluation of the results.

4. RESULTS AND ANALYSIS

4.1 EURUSD AI Agent Result

4.1.1 EURUSD Tear Sheet

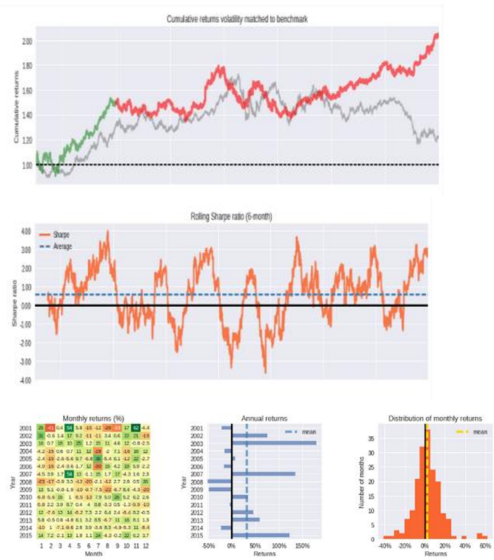

Figure 4. shows the cumulative returns of the AI agent vs. Benchmark (buy-and-hold for EURUSD). It is clearly shown that the AI outperforms buy-and-hold. The Sharpe ratio rolling average for 6 months is 0.7. and the majority of monthly returns show a positive result.

4.1.2 Hypothesis Testing For EURUSD_Agent

4.1.2.1 AI Agent vs. buy-and-hold using annual returns to test the hypothesis

H0: The AI agent's performance is not superior to buy-and-hold performance.

H1: The AI agent's performance is superior to buy-and-hold performance.

Table 1 Paired t-test results for the EURUSD AI Agent vs. buy-and-hold using annual returns data

t-Test: Paired Two Sample for Means		
	Annual Returns_agent	Annual Returns_B&H
Mean	43.88866667	1.466
Variance	5056.348212	108.3599257
Observations	15	15
Pearson Correlation	0.477950253	
Hypothesized Mean Difference	0	
df	14	
t Stat	2.461020542	
P(T<=t) one-tail	0.013727607	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	0.027455215	
t Critical two-tail	2.144786688	

From the above table, the annual returns mean of the AI Agent is 43.88 (variance = 5056. 34) while the annual returns mean of buy-and-hold is 1.46 (variance =108.35). These two annual returns have a positive correlation (0. 47). There is a significant difference between the annual returns of the agent and buy-and-hold, such that the annual returns of the AI agent are superior to the annual returns of buy-and-hold *(P (T<=t) one-tail=0.013, $p < 0.05$).*

Result: The AI agent's performance is significantly superior to the buy-and-hold performance

4.1.2.2 AI Agent vs. CTA (experienced trader) using annual returns to test the hypothesis

H0: The AI agent's performance is not superior to CTA's performance.

H1: The AI agent's performance is superior to CTA's performance.

Table 2 Paired t-test results for the EURUSD AI Agent vs. CTA using annual returns data

t-Test: Paired Two Sample for Means		
	Annual Returns	Annual Returns_CTA
Mean	43.88866667	3.934666667
Variance	5056.348212	28.88141238
Observations	15	15
Pearson Correlation	-0.035775111	
Hypothesized Mean Difference	0	
df	14	
t Stat	2.164144073	
P(T<=t) one-tail	0.024114189	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	0.048228379	
t Critical two-tail	2.144786688	

From the above table, the mean annual returns of the AI Agent is 43.88(variance = 5056.34) while the mean annual returns of CTA is 3.93 (variance =28.88). These two annual returns have a negative correlation (-0.035). There is a significant difference between the annual returns of the AI agent and the CTA, such that the annual returns of the Agent are superior to the annual returns of CTA *(P (T<=t) one-tail=0.024, $p < 0.05$).*

Result: The AI agent's performance is significantly superior to CTA's performance.

Summary of AI agent learning to trade EURUSD

1. The AI agent's performance is significantly superior to buy-and-hold performance.
2. The AI agent's performance is significantly superior to CTA's performance.

4.2 USDJPY AI Agent Result

4.2.1 USDJPY AI Agent tear sheet

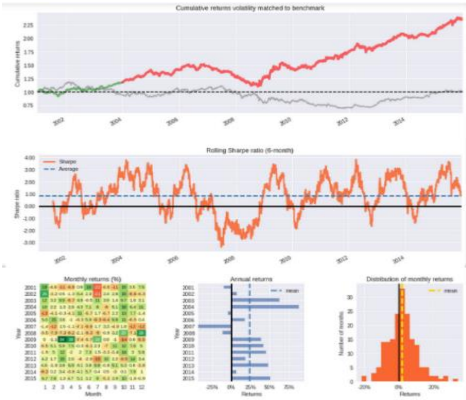

Figure 5 shows the cumulative returns of the AI agent vs. the Benchmark (buy-and-hold for USDJPY). It is clearly shown that the AI outperforms buy-and-hold. The Sharpe ratio rolling average for 6 months is 0.87. Monthly returns mostly show a positive result.

4.2.2 Hypothesis Testing for USDJPY_Agent

4.2.2.1 AI Agent vs. buy-and-hold using annual returns to test the hypothesis

H0: The AI agent's performance is not superior to buy-and-hold performance.

H1: The AI agent's performance is superior to buy-and-hold performance.

Table 3 Paired t-test results for the USDJPY AI Agent vs. buy-and-hold using annual returns data.

t-Test: Paired Two Sample for Means		
	Annual Returns_agent	Annual Returns_B&H
Mean	26.732	0.925333333
Variance	2255.99946	142.8156552
Observations	15	15
Pearson Correlation	0.076078354	
Hypothesized Mean Difference	0	
df	14	
t Stat	2.078459449	
P(T<=t) one-tail	0.028269352	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	0.056538704	
t Critical two-tail	2.144786688	

From the above table, the mean annual return of the AI Agent is 26.73 (variance = 2255.99) while the mean annual return of buy-and-hold is 0.92 (variance = 142.81). These annual returns have a positive correlation (0.07). There is a significant difference between the annual returns of the AI Agent and buy-and-hold, such that the annual returns of the AI Agent are superior to the annual returns of buy-and-hold (*P (T<=t) one-tail=0.028, p* < 0.05).

Result: The AI agent's performance is significantly superior to buy-and-hold performance.

4.2.2.2 AI Agent vs. CTA (experienced trader) using annual returns to test the hypothesis

H0: The AI agent's performance is not superior to CTA' s performance.

H1: The AI agent's performance is superior to CTA' s performance.

Table 4 Paired t-test results for the USDJPY AI Agent vs. CTA using annual returns data.

t-Test: Paired Two Sample for Means		
	Annual Returns_agent	Annual Returns_CTA
Mean	26.732	3.934666667
Variance	2255.99946	28.88141238
Observations	15	15
Pearson Correlation	-0.474885183	
Hypothesized Mean Difference	0	
df	14	
t Stat	1.756304525	
P(T<=t) one-tail	0.0504389	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	0.1008778	
t Critical two-tail	2.144786688	

From the above table, the mean annual returns of the AI Agent are 26.73 (variance = 2255.99) while mean annual returns of CTA are 3.93 (variance =28.88). These two annual returns have a negative correlation (-0.47). There is not significant that the annual of the agent is superior to the annual return of CTA *(P (T<=t) one-tail=0.0504, p > 0.05).*

Result: There is no significant difference between the AI agent's performance and CTA's performance.

Summary of AI agent learn to trade USDJPY

1. The AI agent's performance is significantly superior to buy-and-hold performance.
2. The AI agent's performance is not significantly superior to CTA's performance.

5. DISCUSSION & FINDINGS

The main assumption of this study was that, if there is a pattern in the data, a machine or AI should be able to detect the underlying pattern and make a trading decision better than a human expert, who is believed to be vulnerable to bias, from their own experience and knowledge. The key to understanding the models that we used to test the market depends on the following factors:-

1) The Deep Learning Algorithm

In this paper, we explored an application of DQN (Deep Q Learning), which is one approach of reinforcement learning. There are several parameters related to DQN that determine the performance of our algorithm. For example, different ratios between the test and training sets can show different performance. More training data means the AI can learn several more patterns and can adapt more easily in several trading environments. If the data that we use to train the AI and the data we use to test the AI have the same patterns, it is more likely that the performance will be better than when training with different patterns, and it is vulnerable to curve fitting. We would suggest using as much training data as possible to cover all market modes.

2) Mapping Trading Problems To A Reinforcement Learning Problem

When we map the trading problem to create a reinforcement problem, we need to select the states which determine what the AI will see in the environment or perceive as the world. We must subjectively choose the indicators which we believe could potentially detect some profitable patterns.

Different reward functions could also result in different performance. If we choose the reward function as winning rate or reward to risk, we could train the AI with a more finely tuned objective. For example, if we want to take more risk with the expectation of higher returns, we could set reward to risk as the reward function, to win high profit but with a lower winning rate. However, our Sharpe ratio may possibly be lower. If we want to be more conservative with risk, we could use the winning rate as the reward function, to win more frequently with smaller profits.

3) Deep Neural Network Architecture

The architecture of the deep neural network also contributes to the performance of the AI because the deep network is used as the function approximation to update the weight of each node after calculation of the loss function. A small brain will typically result in lower performance compared to a bigger brain with more hidden layers. However, a non-complex problem such as a predictable pattern will show no difference between a small or big brain. For large-complex problems, bigger brains tend to be better.

4) Trading Objective

Due to different objectives, an AI can be trained using different reward functions such as maximized

Sharpe ratio, profits, winning percentage, reward to risk ratio, annual returns, etc. When we change the reward function, the AI performance will change as well, according to the reward function.

The findings for the EURUSD AI agent vs. buy-and-hold showed that the AI agent significantly outperforms buy-and-hold when using annual returns as the reward. This is an indicator that AI can learn to trade from the data. If we look at the benchmark which is buy-and-hold, it can be seen that if we hold the EURUSD for longer than ten years, we will get almost nothing. This is due to the nature of fiat currency, which is not suitable to be used in the investment class. We would suggest that trading by AI would be better than holding the currency.

The findings for the EURUSD AI agent vs. CTA showed that the AI agent significantly outperforms CTA (experienced trader) when using annual returns as the reward. It does not mean that the AI agent is undoubtedly better than a human expert. The significant differences between the machine and human are caused by emotion. AI can execute a trade without the emotions of fear or greed. When AI detects the profitable pattern, it will not hesitate to take action. Therefore, we would suggest that trading by AI would be better if we care more about annual returns.

The findings for the USDJPY AI agent vs. buy-and-hold showed that the AI agent significantly outperforms buy-and-hold when using annual returns as the reward. This is an indicator that AI can learn to trade from the data. If we look at the benchmark which is buy-and-hold, it can be seen that if we hold the USDJPY for longer than ten years, we will get a slight loss, not mentioning inflation rate. This is due to the nature of fiat currency that is not suitable to be used in the investment class. We would suggest that trading by AI would be better than holding the currency.

The findings for the USDJPY AI agent vs. CTA showed that the AI agent does not significantly outperform CTA (experienced trader) when using annual returns as the reward. In this case, we cannot be sure that the AI is better than a human expert when we compare the returns, even though the mean returns of the AI are better than CTA. However, the standard deviation is also very much higher. Therefore, we would suggest that we need more data to test this hypothesis again. We cannot suggest which one is better, over another.

6. CONCLUSION, LIMITATION & FUTURE RESEARCH

6.1 Conclusion

This study makes several contributions to academics, such as the application of artificial intelligence in algorithmic trading systems development. It is a desirable method to replace the human- decision-making system because the computer can read hidden profitable price patterns better than a human and a computer can execute the trade swiftly and accurately, compared to a human who tends to perform a suboptimal decision-making process when they trade; the AI is the best candidate to replace humans in this situation. Academically, more studies can be conducted to compare the performance of humans and AI.

Moreover, this study supports the opponents of EMH, in that it is possible to develop a trading system to outperform buy-and-hold in the long run.

From this study, we can find a new and alternative method to create return streams that have a low correlation to each other using an AI-generated trading system. As we can see from the results, such as the annual returns of the AI agent and CTA. There is a very low correlation (-0.03) but we could create the min-correlation, risk-diversified portfolio for stable returns. If we can add more return streams with low correlation, we can increase the Sharpe ratio. We can use several AIs to create several return streams that are not correlated with each other.

6.2 Limitation

1. Available Data

To train AI, we need a huge amount of data to learn how to trade. We could not access valuable data such as the actual volume and order of flow between interbank orders. Those data are expensive and available only to giant hedge funds or quant firms.

2. Computing Power

Training the deep neural network is quite expensive in that it consumes time and computing power for complicated calculation. Typically, a bigger brain with more hidden layers would be able to detect more complex patterns and perform complex computation.

3. Trading Assumption

Even though AI performance is entirely satisfactory, it does not mean that we should jump into trading with real money, as the study has only ignited the possibility that we can train AI to trade live in the future. However, when trade lives, we should be aware of how to set up a risk management system to protect from unexpected events such as gap opening/ central bank intervention, nonfarm payroll, news, etc.

6.3 Future Research

1. In future research, we hope that the computing power will be available for training deep neural networks with lower cost. If it is available, the possibilities for trying something new is endless.
2. We could try all possible states. We could input several thousand indicators and more fundamental data. Moreover, more complex cyclic and time series analysis will be added

on to test the model such as singular spectrum analysis to decompose the price series into the cycle.
3. We could try to add some filters such as a hidden Markov model. We could separately train another model to extract market mode only. Hidden Markov model will help us to identify the satisfactory market situation for each specific trading strategy.
4. We could use a bigger brain for the AI. We can add more layers for the AI to increase its capability to learn from the data.
5. We could combine several AIs to become super AI for the portfolio. We could train AIs separately to identify what market AI is best for, to identify what market mode AI is best for, and to identify the correlation between all AIs.
6. We could extend future research by making real live trades with some predetermined risk parameters, such as risk per trade, adding stop loss, adding more advanced pending order, and adding more scale in/scale out algorithms to teach the AI to learn a more complicated trading process.

REFERENCE

Alonso-González, A., Peris-Ortiz, M., & Almenar-Llongo, V. (2015). Providing Empirical Evidence from Forex Autotrading to Contradict the Efficient Market Hypothesis *New Challenges in Entrepreneurship and Finance* (pp. 71-85): Springer.

BarclayHedge, L. (2017). Barclay Currency Traders Index. from https://www.barclayhedge.com/research/indices/cta/sub/curr.html

Chan, K. C., & Teong, F. K. (1995). *Enhancing technical analysis in the Forex market using neural networks.* Paper presented at the Neural Networks, 1995. Proceedings., IEEE International Conference on.

Dempster, M. A. H., & Leemans, V. (2006). An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications, 30*(3), 543-552. doi: 10.1016/j.eswa.2005.10.012

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems, 28*(3), 653-664.

Dourra, H., & Siy, P. (2002). Investment using technical analysis and fuzzy logic. *Fuzzy sets and systems, 127*(2), 221-240.

Du, X., Zhai, J., & Lv, K. (2016). Algorithm Trading using Q-Learning and Recurrent Reinforcement Learning. *positions, 1*, 1.

Dukascopy. (2017). Historical Data Feed. from https://www.dukascopy.com/swiss/english/marketwatch/historical

Ertel, W. (2018). *Introduction to artificial intelligence*: Springer.

Fama, E. F. (1995). Random walks in stock market prices. *Financial analysts journal, 51*(1), 75-80.

Gold, C. (2003). *FX trading via recurrent reinforcement learning.* Paper presented at the Computational Intelligence for Financial Engineering, 2003. Proceedings. 2003 IEEE International Conference on.

Hsieh, D. A. (1991). Chaos and nonlinear dynamics: application to financial markets. *The journal of Finance, 46*(5), 1839-1877.

Huang, S.-M., Hung, Y.-C., & Yen, D. C. (2005). A study on decision factors in adopting an online stock trading system by brokers in Taiwan. *Decision Support Systems, 40*(2), 315-328.

Kalmus, L. P., Trojan, D. R., Mott, B., & Strampfer, J. (1987). Automated securities trading system: Google Patents.

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *The journal of Finance, 55*(4), 1705-1765.

Lu, T.-H., Shiu, Y.-M., & Liu, T.-C. (2012). Profitable candlestick trading strategies—The evidence from a new perspective. *Review of Financial Economics, 21*(2), 63-68.

Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance, 25*(2), 383-417.

Merold, M., Malkin, S., Riordan, T., & Howorka, E. (2002). Automated trading system: Google Patents.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Trans Neural Netw, 12*(4), 875-889.

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting, 17*(56), 441-470.

Nasdaq. (2018). Retrieved 04/21/18, 2018, from https://www.nasdaq.com/investing/glossary/c/commodity-trading-advisor

Neely, C., Weller, P., & Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of financial and Quantitative Analysis, 32*(4), 405-426.

Taylor, M. P., & Allen, H. (1992). The use of technical analysis in the foreign exchange market. *Journal of international Money and Finance, 11*(3), 304-314.

Wang, Y., Wang, D., Zhang, S., Feng, Y., Li, S., & Zhou, Q. (2016). Deep Q-trading.

Whiteson, S. (2010). *Adaptive Representations for Reinforcement Learning* (Vol. 291).

Yao, J., & Tan, C. L. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing, 34*(1), 79-98.