2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

High Frequency Trading with Deep Reinforcement Learning Agents Under a Directional Changes Sampling Framework

George Rayment, Michael Kampouridis
School of Computer Science and Electronic Engineering
University of Essex
Wivenhoe Park, United Kingdom
{gr17754, mkampo}@essex.ac.uk

***Abstract*—High frequency trading strategies in the foreign exchange (FX) market often attempt to extract the latent signals in extremely noisy price moves to help inform trading decisions. Due to the fast-paced environments within which these decisions are made, intelligent trading is an impossible task for the human mind. Deep reinforcement learning (DRL) offers human-like intelligence and high speed computation but, due to the noisy nature of the tick data, can be prone to learning sub-optimal policies as a result of misleading feature and reward signals. In this work we use an intrinsic time sampling method referred to as directional changes (DC), which reports information whenever there is a significant change in price. By sampling tick data from nine FX currency pairs for 2250 datasets, we were able to train reinforcement learning (RL) agents using the Proximal Policy Optimisation (PPO) algorithm to identify and trade profitable strategies in high frequency FX environments. The resultant models were compared to four benchmarks including buy and hold, moving average crossover, relative strength index and a rule-based DC strategy, across three different metrics (namely returns, maximum drawdown, and Calmar ratio), with the reinforcement learning models outperforming them all.**

***Index Terms*—directional changes, high frequency trading, machine learning, deep reinforcement learning**

I. Introduction

The foreign exchange (FX) market is a decentralized market that facilitates the trading of currencies twenty-four hours per day and five days per week. Various strategies are used to trade the FX market profitably, but all focus on successfully predicting price movements. For the growing number of market participants that use machine learning (ML) to trade, this generally involves two main steps: identifying significant market indicators and applying the appropriate ML techniques [1]. Indicators depend on the underlying data which can itself be sampled in various ways, including fixed interval sampling and intrinsic time sampling. Fixed interval sampling involves taking a snapshot of the price at fixed time intervals which results in a physical time series of price data. Due to the non-linear and non-stationary nature of FX data, the fixed interval sampling approach is prone to ignoring significant price movements that may occur between adjacent snapshots [2]. Intrinsic sampling is an alternative approach based on taking snapshots of the market when events considered significant occur [3].

Directional changes (DC) sampling [4] is an example of an intrinsic sampling technique. The motivation for using DC in this work is firstly, due to the inherent confirmation of an existing trend, simple rule-based strategies can be built using just the sampling information. Secondly, the nature of the sampling algorithm provides significant information clarity over fixed interval sampling as it reports significant price moves by removing the noise, therefore removing many of the false signals associated with other sampling methods. The DC sampling method generates an event summary by recording key events in the market according to a threshold θ. The threshold, predetermined by a trader, expresses a percentage change in market price that is considered significant. The key events are recorded as alternating upward and downward directional changes trends respectively. Each trend is subdivided into a directional change (DC) event and an overshoot (OS) event with the OS event immediately following the DC event.

ML is a technique that has been used to create profitable trading strategies both with [5] and without [6] the DC framework. DRL enables agents to learn complex policies [7], [8] and make decisions almost instantaneously, due to the policy function being learned by a deep neural network. In this work we leverage PPO, a type of DRL algorithm known for it's training efficiency and stability [9], both of which offer significant benefits when dealing with noisy financial data. We create an environment within which the DRL agent is able to learn and trade using DC data. Our aim is to show that the DC sampling algorithm can provide the agent with the correct information and enables the agent to use it's inherent ability to learn complex policies, to outperform other benchmark strategies. This would therefore demonstrate the value of the combination of both DRL and DC sampling within a high frequency FX environment.

The rest of the paper is organised as follows: Section II presents an overview of the related empirical work that explore ML and RL techniques for trading. Section III presents the background information required for an understanding of the DC framework and deep reinforcement learning. Section IV describes our methodology. Section V presents the experimental setup. In Section VI we present our findings. Finally, Section VII concludes the paper and discusses future work.

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

II. Related Work

In 1997 [4] introduced the concept of transforming physical time series into DC series and empirically formalised 12 DC-based scaling laws using high-frequency data from 13 major FX markets. Subsequent works, such as [10] formalised additional scaling laws. Other works, such as [11] proposed new DC-based indicators, which were used for tasks such as profiling data and identifying regime changes.

ML is a popular technical approach for building DC based trading strategies. Works such as [12] and [13] use evolutionary algorithms to build trading strategies. In both works the DC framework outperformed the physical time-based algorithm. Genetic algorithms (GAs) have also been used to develop strategies in DC event-series [14] and [15]. Both [14] and [15] generate DC based strategies using GAs that outperform a number of non-DC benchmark strategies.

A number of approaches to trading use ML methods other than evolutionary algorithms and without the DC framework such as [6]. These works span multiple financial instruments, demonstrating effective derivations of numerous different trading strategies. Reinforcement learning has been applied to algorithmic trading in plenty of literature [16], [17]. The approaches taken by researchers vary, but ultimately revolve around the design of the environment. The environment itself encapsulates the action space, state space and reward function. This fundamental approach, that can be observed as early as 1998 with [16], remains consistent today with recent papers such as [18], also demonstrating the same emphasis on the careful cultivation of environment representation and achieving similarly promising results.

To the best of our knowledge, only shallow RL techniques have been used in conjunction with the DC sampling framework [19], [20]. Both these works use a system referred to as 'DCRL', that uses Q-learning to identify an optimal trading policy with lookup tables, as opposed to deep neural networks. The performance of the DCRL algorithm demonstrates favourable performance in both works when evaluated on stock market data. The combination of DC event series created from high-frequency FX data and ML techniques, in general, has offered a successful approach to creating trading strategies. However, after conducting the above review of the literature, it can be deduced that DRL has not been explored in depth when considered alongside directional changes. Given the success of DRL in other domains and the portability of this approach to high frequency trading, we propose a novel system of DRL agents for trading in high-frequency FX trading environments under the DC sampling framework.

III. Background Information

A. Directional Changes

Directional changes is a data sampling technique used in creating intrinsic time-series from a physical time-series. First, a threshold value θ that expresses a significant change in price is predetermined by a trader. Successive alternating snapshots of the market are then recorded when a change in price is equal to or greater than the threshold, creating a time-series that obfuscates noise between adjacent snapshots.

A DC trend can either be an uptrend or a downtrend. The DC trend is composed of a directional change (DC) event and an overshoot (OS) event. A directional changes confirmation (DCC) point is the moment in time when price is observed to be greater than a given threshold and demarcates the DC event from the OS event. The end of an OS event is known as a directional change extreme (DCE) point. It is determined in hindsight, after the next DC event in the opposite direction is confirmed. To conduct the sampling algorithm the initial price is considered the DCC point of the first upward move. From this initial DCC point the algorithm iterates over each price, noting any new extreme points as the current DCE of the move and locks this value in as soon as the DCC in the other direction is posted. A graph demonstrating this algorithm in action is shown in Figure 1. The green and red dotted lines represent the DC and OS moves of the 0.025% sampling summary while the blue and purple solid lines represent the DC and OS moves of the 0.015% sampling summary, both are laid over the raw tick prices represented by the thinner solid grey line. Point A denotes the DCE of the previous down trend and the start of the new up trend. This continues to point B which signifies the DCC of the uptrend with the following dotted red line denoting the OS move. From the figure we can see that a lower threshold value means a higher sampling frequency as the numerous moves in period C all occur in the time it takes the 0.025% move from A to D, consisting of a single up and down trend. This constitutes one of the advantages of DC sampling. More specifically, the use of different DC thresholds provides a different view of the data: smaller thresholds allow the detection of more events and, as a result, actions can be taken promptly; on the other hand, larger thresholds detect fewer events, but provide the opportunity of taking actions when bigger price variations are observed.

B. Reinforcement Learning

Reinforcement learning (RL) is a subset of ML that relies on the use of an agent, an environment and the communication of states, actions and rewards to train an optimal policy that maximises reward in an environment. The policy itself is the decision making mechanism of the agent, which generates an action based on the state and reward provided by the environment at the current time step. Taking this action generates a new state and reward and the cycle continues until a termination condition is met. This cycle acts as the agent's experience and different RL algorithms are implemented in order to build out the policy that maximises the cumulative reward over the whole engagement with the environment.

Deep reinforcement learning (DRL) refers to the intersection of reinforcement learning and deep learning, where the reinforcement learning algorithm is designed to train a deep neural network to learn the policy. Since a policy is just a mapping of states to actions, the format of this mapping can be extended from a simple table with learned values (as used in traditional Q-learning techniques [19]) to a deep neural network, as they are universal function approximators. Using a deep neural network to learn the optimal policy to move around the environment is beneficial for learning policies in environments with large, continuous state spaces as optimal actions can be inferred from the experience of similar states.

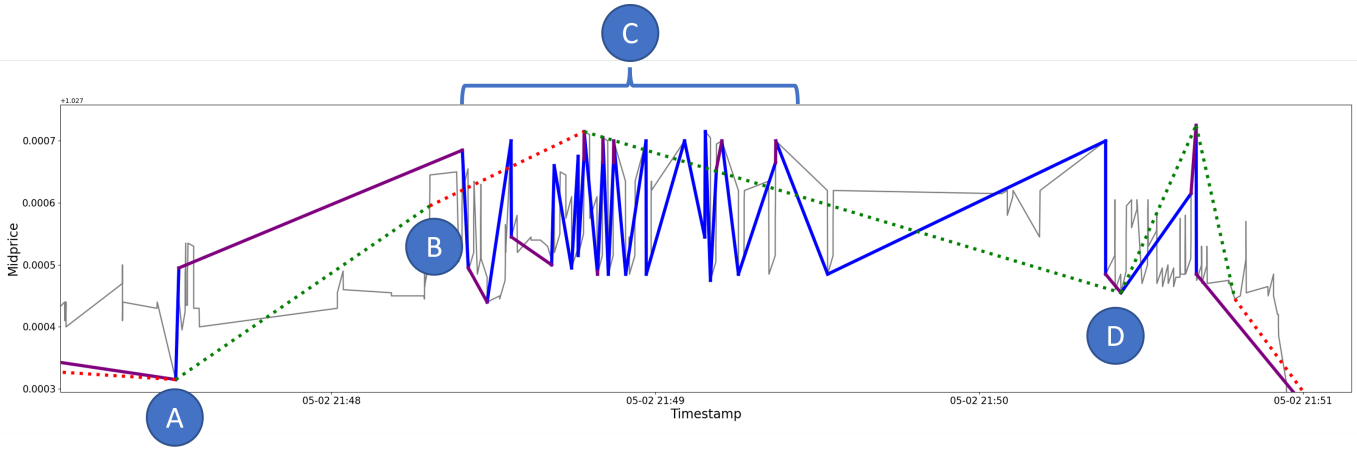

Fig. 1: Directional Changes Sampling Diagram of EUR/CHF at $\theta = 0.015\%, 0.025\%$

Proximal Policy Optimisation (PPO) [9] is a DRL algorithm that has been designed to take smaller steps when optimising the policy in order to make the learning process more stable than other DRL algorithms. A common implementation of PPO uses an actor-critic architecture. Both the actor and critic can be represented as a deep neural network that partially share a beginning portion of the network, but ultimately produce different outputs. The task of the actor network is to learn the policy as a mapping of states to actions and the task of the critic network is to learn the cumulative reward as a result of the states and actions.

The general approach to training using PPO starts with the actor taking actions based on the current policy. The expected cumulative reward is then calculated by the critic network and used in the calculation of the advantage value. Once the advantage is calculated from a batch of time steps, the result of the objective function can be obtained. This objective function then allows the actor-critic network to apply gradient descent to both the actor and critic network and improve the policy. See Algorithm 1 for the pseudo code of this algorithm.

Algorithm 1 PPO Algorithm

Require: Actor policy π_θ with parameters θ, environment with maximum time steps T, number of iterations N, number of epochs K

for $i \leftarrow 1$ to N **do**
  Collect a set of trajectories $\mathcal{D} = \tau$ using actions generated by policy π_θ in the environment for T time steps
  Compute advantages $A(\tau)$ for each state in each trajectory using the critic network
  Compute surrogate objective $\mathcal{L}(\theta)$ using the trajectories and advantages
  for $j \leftarrow 1$ to K **do**
    Compute gradients $\nabla_\theta \mathcal{L}(\theta)$ using $\mathcal{D}$ and $A(\tau)$
    Update policy π_θ parameters using optimiser
  end for
end for

IV. Methodology

An overview of the methodology is presented in Figure 2. The data preparation phase (Section IV-A) begins by transforming the tick data into a number of windows, each split into training, validation and test sets, and subsequently transforming the above data from physical tick data to events series and generate the relevant indicators through the application of the DC sampling algorithm. Next, we develop the DRL environment (i.e. action space, state space, and reward function) (Section IV-B), and train the neural network to learn the optimal policy (Section IV-C). Afterwards, the algorithm hyperparameters are tuned on the validation set and we then re-train the model using the newly identified hyperparameter values. In the end, we calculate the performance metrics (presented in Section IV-D) on the test set. We present the methodology in detail next.

A. Data Preparation

The raw tick data is first split into contiguous sets of weekly data. Rolling windows are then generated from groups of four consecutive weeks as shown in Figure 2. Rolling windows allow each model to learn from recent market history, so price behaviour in the training data is similar to the test data [11]. Data split this way allows the formation of contiguous test sets. Each window is then also split into training, validation and test sets before applying the DC sampling framework. The DC indicators in Table I and the start and end price of each DC move are used as features. N_{DC}, C_{DC}, and A_T have 6 period variants: for 1, 10, 20, 30, 40, and 50 DC events. We also use a 3, 5, and 10 events moving average for OSV, T_{DC}, and R_{DC}, as this provides an aggregation of more recent events and has shown to be a successful approach in [13]. Aggregating some of the indicators in this way provides more noise resistant signals to the agent. Indicators with periods of 1 are also used to give the agent some knowledge of recent price history.

B. Environment Development

Each window undergoes the same training process with the same environment definition after the data is prepared. The environment is defined as a custom environment in the

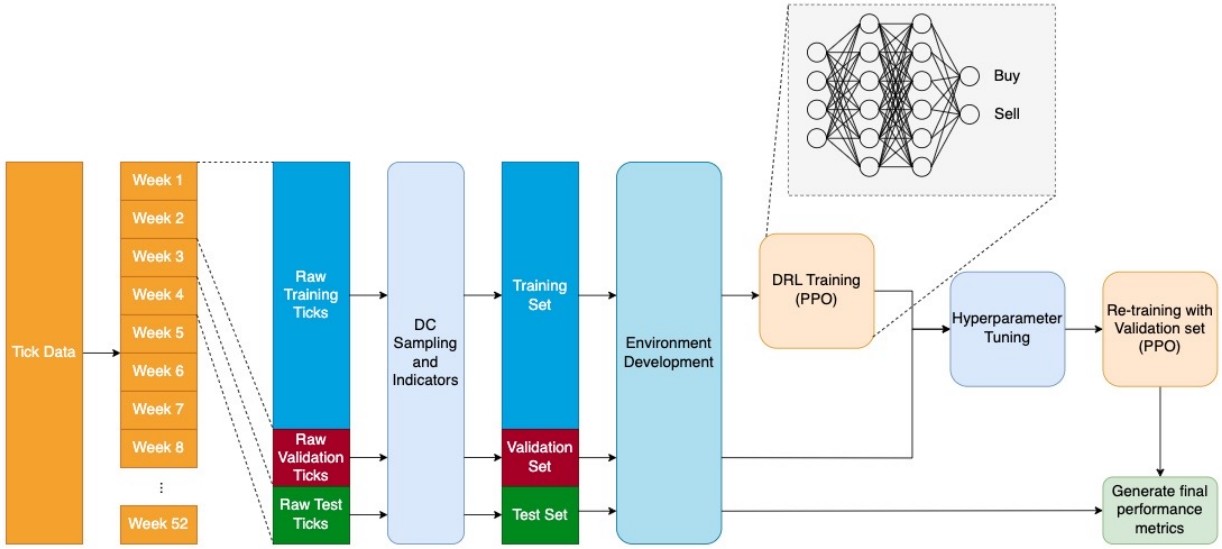

Fig. 2: Experiment Methodology (see Section V-B for real network architecture)

TABLE I: DC Indicators
Where: θ is DC threshold and DCC is DC confirmation point (Periods marked with a * use a moving average of the indicator)

Indicator	Description	Equation	Period
$\boldsymbol{TMV}$	Ratio of whole price move to threshold	$\frac{\lvert\Delta price\rvert}{\theta}$	1
$\boldsymbol{OSV}$	Percentage change between current DCC and previous DCC normalised by threshold	$\frac{\left(\frac{DCC_t - DCC_{t-1}}{DCC_{t-1}}\right)}{\theta}$	(3, 5, 10)*
$\boldsymbol{R_{DC}}$	Number of ticks adjusted by the return of the event	$\frac{(TMV * \theta)}{\Delta Event_t}$	(3, 5, 10)*
$\boldsymbol{T_{DC}}$	Number of ticks over the course of the event	$\Delta Event_{no.ticks}$	(3, 5, 10)*
$\boldsymbol{N_{DC}}$	Number of ticks over a certain number of events	$\sum_{i=0}^{n} Event_{no.ticks_i}$	(1, 10, 20, 30, 40, 50)
$\boldsymbol{C_{DC}}$	Sum of $\lvert TMV\rvert$ over a certain number of events	$\sum_{i=0}^{n} \lvert TMV\rvert_i$	(1, 10, 20, 30, 40, 50)
$\boldsymbol{A_T}$	Difference between the number of ticks spent on an up and down trend over n events	$\sum_{i=0}^{n} UpEvent_{no.ticks_i} - \sum_{i=0}^{n} DownEvent_{no.ticks_i}$	(1, 10, 20, 30, 40, 50)

`Gym` [21] library so that it is compatible with the appropriate reinforcement learning and deep learning libraries.

1) Action Space: The action space of the agent is the set of possible actions the agent can take at any given time step. We choose the discrete action space of buy or sell for this experiment after some preliminary testing. During the preliminary testing we found that, if given the option to buy, sell or hold, the agent holds to avoid transaction costs. By restricting the agent to buy or sell we force the agent to make a trading decision. The agent therefore holds the position until it suggests an action in the opposite direction. This approach appears to provoke much more trading while still allowing for transaction costs and coaxes the agent into learning a more profitable strategy.

2) State Space: State space is the representation of the environment at any given time step. The state at each time step is represented by a preceding window of price data and relevant features as defined in Section IV-A. The selection of the appropriate size for the previous window of time steps (also known as the state space lag) to be provided as input to the agent, represents another hyperparameter that requires tuning (see Section V-B for details).

3) Reward Function: Each agent is provided with a fixed starting balance and is rewarded with the profit of a trade once the position is closed and otherwise receives a reward of zero. This reward function then creates a balance between the agent's desire to trade as much as possible to build up profit but also prompts the agent to identify effective trading opportunities that would not present losses.

C. Deep Reinforcement Learning Training

The training process involves iterating over the environment for a number of time steps, a hyperparameter of the training algorithm defined in Section V-B. With each time step the state space is passed to the deep neural network which in turn produces an action and expected cumulative return. This process is largely abstracted by the `Stable Baselines 3` (SB3) library [22] with a `PyTorch`[1] backend and can automatically step through this process and apply the appropriate adjustments to the weights of the actor-critic deep neural network. This whole process returns a trained model per window, which then uses the validation set to identify the correct set of hyperparameters as shown in Figure 2.

D. Performance Metrics

Trained agents are tested using a number of metrics, all of which are based on marginal return (see Equation 1) which is the return of each individual trade. The performance metrics

[1]https://pytorch.org/

used are Total Return (see Equation 2), Maximum Drawdown (see Equation 3) and Calmar ratio (see Equation 4) [2]. Total Return measures the final return of the system of agents and Maximum Drawdown is used to measure the risk of the system of agents by calculating the losses that have to be incurred in order to produce the final return. Calmar Ratio is then used to represent both these performance metrics in a single value and therefore provide a measure of risk adjusted return.

$$MR = \pm\%\Delta p * P_{size} \tag{1}$$

where: MR is marginal return, p is price, sign is changed to reflect profit and loss, and P_{size} is the position size, this represents the quantity of the currency pair being traded for a specific position.

$$R = \sum_{i=0}^{no.trades} MR_i \tag{2}$$

where: R is total return, and MR_i is the marginal return defined in Equation 1 for trade i.

$$MDD = \frac{\rho - \tau}{\rho} \tag{3}$$

where: MDD is maximum drawdown, ρ is the peak balance before largest drop, this refers to the largest balance observed at the peak of the largest drop in the balance, and τ is next lowest balance before a new high, this refers to the lowest balance observed after the peak balance before the balance rises above the peak value.

$$CalmarRatio = \frac{R}{MDD} \tag{4}$$

E. Trade Filtering

In preliminary result analysis on a subset of pairs and thresholds, the agent seemed to learn to trade in periods of low volatility (see period C in Figure 1). These low volatility periods consist of successive DC trends with little or no overshoot as shown by the 0.015% DC moves in Figure 1. The agent appears to identify these low volatility periods and engage in the fast opening and closing of trading positions by entering a position at the DCC point of an up or down trend in the opposite direction and exiting the trade on the next tick. This behaviour continues until the DRL agent identifies the end of the low volatility period. As a consequence of these observations, a filter is added that only allows the agent to trade when there has been a series of trends with no overshoot. The optimal number of preceding no-overshoot trends is identified using a grid search as described in Section V-B.

V. Experimental Setup

A. Data

The tick data for the nine currency pairs is downloaded from TrueFX.com[3]. Data for currency pairs AUD/JPY, EUR/CHF, EUR/GBP and EUR/JPY are taken from the period 01/05/2022 to 30/04/2023, the remaining five currency pairs all include USD (AUD/USD, GBP/USD, NZD/USD, USD/CHF and USD/JPY) and are sampled from the period of 01/02/2021 to 31/01/2022. After some preliminary tested we decided that each window consists of 4 weeks of data with a shift of 1 week between windows as this generates an appropriate number of trends per window suitable for training. The sampling process is applied independently per training, validation and test set and is repeated for the set of windows per pair with an array of seven DC thresholds (θ), ranging from 0.017% to 0.025% with gaps of 0.002% between each threshold totalling 2250 (9 pairs $\times$ 5 thresholds $\times$ 50 windows) datasets. Each threshold produces a different summary of the data, we therefore need to test the DRL strategy on multiple thresholds to identify how effective it is at trading under the DC sampling framework.

B. Hyperparameter Tuning

An approach is developed to filter out trades suggested by the agent outside low volatility periods. Using the DC framework, the filter identifies how many previous trends have no overshoot event. If the number of previous consecutive no OS trends is above a certain threshold then the agent is allowed to trade. The correct value was found with a grid search over values of 2, 5, 10 and 20 on the validation set, with the best performing value used to produce the final results from the test set. We also optimise state space lag (see Section IV-B2) and training time steps (see Section IV-C) using a grid search on a subset of all pair and threshold data. State space lag was optimised over values of 5, 10 and 20 and training time steps was grid searching up to 1 million time steps. We found a performance drop off for larger state space lag values so 5 trends was the most appropriate value. We also found that models performed most favourably after 200,000 time steps of training with no performance gains after this point.

We ran a grid search over an array of different deep neural network model architectures and activation functions for the PPO policy and value networks. Using a subset of all validation sets, we determined that with two hidden layers of 64 units each and the ReLU activation function we obtained the best results. The number of input neurons is defined by the 30 input features per time step, as shown in Table I and the DC start and end prices, multiplied by the state space lag of 5 time steps. The final two neurons of the policy network are then buy and sell actions. The full architecture of the policy network is therefore (150, 64, 64, 2).

C. Benchmarks

The following benchmarks have been devised with the intention of testing how the DRL agent compares to commonly-used trading strategies.

a) Buy and Hold (B&H): The B&H strategy enters a long position on the first trend and then exits that position on the final trend, making a single trade over the duration of the data. The B&H strategy is a common financial benchmark, as it's a passive strategy (not active trading) and is a useful comparison to strategies that perform active trading.

[2]The Calmar ratio metric was used to focus more on the worst case scenario, a topic more appropriate for high-frequency traders, as opposed to Sharpe Ratio as mentioned by Richard Olsen in https://hughchristensen.com/papers/academic_papers/eforex-072007.pdf

[3]https://www.truefx.com/truefx-historical-downloads/

b) Moving Average Crossover (MAC): This is technical analysis strategy that uses moving average (MA) crossover signals. This strategy calculates three MAs at periods of 7000, 14000 and 28000. The slowest MA (28000 period) is used as a trade filter, if the 7000 period MA crosses above the 14000 period MA and this happens above the filter MA, then a buy trade is entered. Conversely, if the 14000 period MA crosses over the 7000 period MA underneath the filter MA then a sell signal is given and the trade is exited. Factors of these three MA period choices are commonly used in the literature [23].

c) Relative Strength Index (RSI): The RSI strategy is based on the RSI technical analysis indicator. The RSI strategy uses the RSI indicator with a 140 period and values of 75 and 25 as overbought and oversold levels. The 140 period was selected based on the period choices in the literature [23] and a grid search was run over overbought and oversold levels to determine the best pair of values. The entry rules are defined so that whenever the RSI crosses below the oversold level, a buy trade is executed and the opposite when the overbought level is crossed. As the price reverses back to the market equilibrium and reaches the opposite overbought or oversold level, the trade is exited and any profits or losses are taken.

d) Blind Low Volatility (BLV): This benchmark strategy uses the low volatility filter applied to the DRL agent but with a simple rule-based trading strategy that tells the agent to buy at the DCC of a downtrend and sell at the DCC of an uptrend when the filter allows trading. When the filter no longer allows trading, the rule-based system will exit the market and not trade until permission from the filter is granted again. This system is used as a comparison to test if the DRL agent is learning a simple rule-based strategy or if there is more intelligent behaviour being exhibited.

VI. Results

The following results show the outcome of simulating trading for a whole year across the test set of each window, per pair, under a 0.025% transaction cost.

The return results in Table II show that there are no pairs with extreme losses and the pairs that make gains tend to make large gains. Pairs EUR/CHF, EUR/GBP and EUR/JPY demonstrate extremely high profit levels at all thresholds. When analysing the trading decisions made for these pairs it is clear that the agent has learnt to trade quickly during prolonged periods of low volatility. This rapid entry and exit of trades demonstrates that the agents have identified profitable periods within the price data across many different training sets and applied this policy profitably to the test set. AUD/JPY and USD/CHF also performed well with positive returns across all thresholds. The remaining pairs produced returns ranging from -1.80% to 0.07. Given the lack of low volatility periods in the other pairs it can be deduced that the agents tend towards learning these quick entry and exit strategies, as demonstrated in EUR/CHF, EUR/GBP and EUR/JPY, that make significant profits. It is also worth noting that DRL's performance is consistently good across all DC thresholds, and it improves as the DC threshold increases. Lastly, the benchmark strategies all produced marginally positive and negative returns under the same position sizes and transaction costs, apart from the buy and hold strategy which consisted of larger and more sporadic positive and negative returns.

The associated risk with the trading strategies was also measured using maximum drawdown as defined in Equation 3. The results in Table III show that all pairs generate comparable levels of maximum drawdown. NZD/USD demonstrates the highest levels of maximum drawdown across all pairs, this poor performance is consistent with the total return results as NZD/USD also performed the worst for each threshold. Most other pairs produce maximum drawdown levels of less than 1% with only a few just exceeding this value. This is most likely due to the low position sizes of the trades, allowing models to excel in periods of low volatility where multiple short trades can be entered in quick succession. When comparing the DRL algorithm to the traditional technical analysis benchmark strategies of MAC and RSI, it is clear that there are no differences in maximum drawdown of any magnitude. The BLV strategy, demonstrates significant levels of maximum drawdown reaching highs of around 80% in some cases.

We can measure the risk adjusted return of each strategy by using the Calmar ratio (Equation 4) as this is an aggregated measure of total return and maximum drawdown. The results in Table IV show that the Calmar Ratio follows a similar pattern to the total returns in Table II. EUR/CHF, EUR/GBP and EUR/JPY are the best performers with extremely high Calmar Ratios, AUD/JPY performed well with results again following the same pattern as the total returns. USD/CHF showed Calmar Ratios above 1.0 [4] on all but one threshold, again demonstrating strong performance. We also see DRL's performance improving for higher DC thresholds. RSI also showed some good results but not to the level of the DRL strategy. RSI is a conservative strategy by nature and as a result can provide some consistently positive results with favourable maximum drawdown levels. The risk/reward ratio however, as shown by the Calmar Ratio, exposes the downsides of this RSI strategy as it performs worse than the DRL strategy despite the favourable maximum drawdown results. The MAC strategy was outperformed by both the RSI strategy and the DRL strategy but outperformed the BLV strategy. It can also be observed from Tables II, III and IV, that the larger thresholds tend to perform better than lower thresholds.

To further investigate the algorithms' returns performance, we applied Friedman's non-parametric test, and we present the results in Table V. For each algorithm, the table shows the average rank according to the Friedman test (first column), and the adjusted p-value of the statistical test when that algorithm's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to the Conover post-hoc test (second column).[5] As we can observe, the proposed DRL algorithm ranks first in terms of returns and Calmar ratio, and statistically outperforms all

[4] A Calmar Ratio above 1.0 means that total return is greater than maximum drawdown.

[5] Due to the fact that the B&H, MAC, and RSI methods are not using the DC framework, we can only obtain a single metric value (return, maximum drawdown, Calmar ratio) per currency pair, whereas for DRL and BLV we obtain 5 values (one per threshold). To be able to perform the statistical tests, the B&H, MAC, and RSI values are duplicated to match the number of thresholds per pair.

TABLE II: Total Return (%) by DC threshold for DRL and BLV. B&H, MAC, and RSI strategies are physical time based strategies, so only a single value is presented per currency pair. Best value per currency pair is denoted in boldface.

Pair / Treshold	0.017%		0.019%		0.021%		0.023%		0.025%		B&H	MAC	RSI
	DRL	BLV	DRL	BLV	DRL	BLV	DRL	BLV	DRL	BLV			
AUD/JPY	8.21	-33.42	14.17	-28.85	20.92	-24.92	22.78	-22.26	**24.02**	-19.87	-0.13	-2.02	1.04
AUD/USD	**0.07**	-0.93	-1.11	-0.78	-0.69	-0.62	-0.76	-0.56	-1.09	-0.38	-10.48	-0.33	-0.15
EUR/CHF	858.26	-86.25	1830.56	-80.36	1848.58	-75.17	**2788.97**	-70.80	2518.33	-67.23	-4.41	-1.55	0.95
EUR/GBP	619.09	-78.84	834.59	-73.10	752.50	-67.50	996.22	-63.16	**1025.83**	-58.47	3.85	-1.48	-0.26
EUR/JPY	5698.05	-95.57	8007.05	-93.50	10196.62	-91.30	**11586.09**	-88.99	9795.96	-86.61	10.95	-2.92	1.50
GBP/USD	-0.23	-4.16	-0.88	-3.18	-0.08	-2.14	-0.07	-1.65	-0.84	-1.24	-3.86	-0.41	**1.22**
NZD/USD	-1.80	-2.68	-1.14	-2.28	-1.66	-2.09	-0.76	-1.82	-1.24	-1.77	-10.01	-0.30	**0.31**
USD/CHF	0.46	-4.71	0.26	-4.18	1.06	-3.44	1.14	-3.08	1.42	-2.81	**2.73**	-0.18	-0.43
USD/JPY	-0.12	-2.72	-0.22	-2.14	-0.28	-1.57	-0.38	-1.33	-0.27	-1.14	**8.96**	-0.18	0.21

TABLE III: Maximum Drawdown (%) by DC threshold for DRL and BLV. MAC and RSI strategies are physical time based strategies, so only a single value is presented per currency pair. B&H cannot be calculated, as it only performs a single trade. Best value per currency pair is denoted in boldface.

Pair / Treshold	0.017%		0.019%		0.021%		0.023%		0.025%		MAC	RSI
	DRL	BLV	DRL	BLV	DRL	BLV	DRL	BLV	DRL	BLV		
AUD/JPY	0.95	33.42	1.21	28.85	0.52	24.92	0.63	22.26	0.44	19.87	2.03	**0.45**
AUD/USD	0.45	0.93	1.25	0.78	0.71	0.63	0.96	0.57	1.18	0.38	**0.35**	0.57
EUR/CHF	0.88	86.25	0.11	80.36	0.13	75.17	0.09	70.80	**0.04**	67.23	1.55	0.16
EUR/GBP	0.48	78.84	0.12	73.10	0.09	67.50	**0.05**	63.16	0.10	58.47	1.49	0.77
EUR/JPY	0.19	95.57	0.15	93.50	0.15	91.30	**0.14**	88.99	0.21	86.61	2.92	0.30
GBP/USD	0.36	4.16	1.21	3.18	0.50	2.14	0.39	1.65	0.92	1.24	0.41	**0.27**
NZD/USD	1.83	2.68	1.53	2.28	1.81	2.09	1.09	1.82	1.38	1.77	**0.30**	0.60
USD/CHF	0.35	4.71	0.35	4.18	0.58	3.44	0.26	3.08	0.28	2.81	**0.19**	0.78
USD/JPY	0.30	2.72	0.65	2.14	0.34	1.57	0.48	1.33	0.32	1.14	**0.20**	0.43

TABLE IV: Calmar Ratio (%) by DC threshold for DRL and BLV. MAC and RSI strategies are physical time based strategies, so only a single value is presented per currency pair. B&H cannot be calculated, as it only performs a single trade. Best value per currency pair is denoted in boldface.

Pair / Treshold	0.017%		0.019%		0.021%		0.023%		0.025%		MAC	RSI
	DRL	BLV	DRL	BLV	DRL	BLV	DRL	BLV	DRL	BLV		
AUD/JPY	8.64	-1.00	11.71	-1.00	40.23	-1.00	36.16	-1.00	**54.59**	-1.00	-1.00	2.31
AUD/USD	**0.16**	-1.00	-0.89	-1.00	-0.97	-0.98	-0.79	-0.98	-0.92	-1.00	-0.94	-0.26
EUR/CHF	975.30	-1.00	16641.45	-1.00	14219.85	-1.00	30988.56	-1.00	**62958.25**	-1.00	-1.00	5.94
EUR/GBP	1289.77	-1.00	6954.92	-1.00	8361.11	-1.00	**19924.40**	-1.00	10258.30	-1.00	-0.99	-0.34
EUR/JPY	29989.74	-1.00	53380.33	-1.00	67977.47	-1.00	**82757.79**	-1.00	46647.43	-1.00	-1.00	5.00
GBP/USD	-0.64	-1.00	-0.73	-1.00	-0.16	-1.00	-0.18	-1.00	-0.91	-1.00	-1.00	**4.52**
NZD/USD	-0.98	-1.00	-0.75	-1.00	-0.92	-1.00	-0.70	-1.00	-0.90	-1.00	-1.00	**0.52**
USD/CHF	1.31	-1.00	0.74	-1.00	1.83	-1.00	4.38	-1.00	**5.07**	-1.00	-0.95	-0.55
USD/JPY	-0.40	-1.00	-0.34	-1.00	-0.82	-1.00	-0.79	-1.00	-0.84	-1.00	-0.90	**0.49**

algorithms but RSI. In terms of maximum drawdown, DRL ranks marginally second after RSI (average rank 2.03 vs 2.06), but is not statistically outperformed.

From the above, we can conclude that the proposed DRL algorithm performs strongly in terms of returns and Calmar ratio. In terms of risk, it's performance is on par with RSI and MAC. However, due to the fact that DRL produces considerably higher returns for a similar level of risk on the pairs that RSI also performs well on, we can conclude that DRL is the best performing strategy overall. From Table IV we can can see that when DRL is outperformed by RSI, it is often in the order of 0.1%, whereas when DRL outperforms RSI it is at a considerably higher magnitude, which is a factor the significance testing does not account for.

VII. Conclusion

In conclusion, this paper has demonstrated that the combination of DRL with the directional changes framework leads to profitable results at reduced risk. Our results have also shown that the proposed DRL algorithm learns a strategy that is able to outperform well-known physical time based strategies such as B&H, MAC, and RSI. The strategy learned often consists of short successive trades that last for short periods of time.

The approach of trading in low volatility periods has worked well for the agents in this experiment with a realistic fixed transaction cost of 0.025%. In true market conditions the spread can be variable and potentially widen to levels higher than our fixed transaction cost with lower price volatility. This motivates us in future work to train separate, spread-aware DRL agents in exclusively high volatility periods. These periods offer more profit potential per trade and allow trades to be simulated closer to the real market. This could lead to more advanced systems that involve invoking agents that are more appropriate given the market regime.

References

[1] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, "Forecasting stock market crisis events using deep and statistical machine learning techniques," *Expert systems with applications*, vol. 112, pp. 353–371, 2018.

[2] Y. Tang, Z. Song, Y. Zhu, H. Yuan, M. Hou, J. Ji, C. Tang, and J. Li, "A survey on machine learning models for financial time series forecasting," *Neurocomputing*, vol. 512, pp. 363–380, 2022.

[3] B. Mandelbrot and H. M. Taylor, "On the distribution of stock price differences," *Operations research*, vol. 15, no. 6, pp. 1057–1062, 1967.

TABLE V: Statistical test results for return (left), maximum drawdown (middle), and Calmar ratio (right), according to the non-parametric Friedman test with the Conover's post-hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface. B&H is only included in the returns table, as it only performs a single complete trade (buy on the first day and sell on the last), and as a result maximum drawdown and consequently Calmar ratio cannot be defined.

(a) Returns

Friedman test p-value		**1.53e-22**
	Ave. Rank	p_{Con}
DRL (c)	2.02	-
RSI	2.14	2.04e-1
B&H	**3.08**	**1.57e-6**
MAC	**3.16**	**2.53e-13**
BLV	**4.60**	**2.23e-34**

(b) Maximum Drawdown

Friedman test p-value		**1.55e-19**
	Ave. Rank	p_{Con}
RSI (c)	2.03	-
DRL	2.06	9.50e-1
MAC	2.06	5.16e-2
BLV	**3.84**	**7.43e-25**

(c) Calmar ratio

Friedman test p-value		**2.53e-36**
	Ave. Rank	p_{Con}
DRL (c)	1.44	-
RSI	1.57	4.19e-1
MAC	**2.98**	**1.16e-46**
BLV	**4.00**	**9.98e-83**

[4] D. M. Guillaume, M. M. Dacorogna, R. R. Davé, U. A. Müller, R. B. Olsen, and O. V. Pictet, "From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets," *Finance and stochastics*, vol. 1, no. 2, pp. 95–129, 1997.

[5] A. Adegboye, M. Kampouridis, and F. Otero, "Algorithmic trading with directional changes," *Artificial Intelligence Review*, pp. 1–26, 2022.

[6] S. Lahmiri and S. Bekiros, "Intelligent forecasting with machine learning trading systems in chaotic intraday bitcoin market," *Chaos, Solitons & Fractals*, vol. 133, p. 109641, 2020.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[10] J. B. Glattfelder, A. Dupuis, and R. B. Olsen, "Patterns in high-frequency fx data: discovery of 12 empirical scaling laws," *Quantitative Finance*, vol. 11, no. 4, pp. 599–614, 2011.

[11] E. Tsang and J. Chen, "Regime change detection using directional change indicators in the foreign exchange market to chart brexit," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 3, pp. 185–193, 2018.

[12] J. Gypteau, F. E. Otero, and M. Kampouridis, "Generating directional change based trading strategies with genetic programming," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2015, pp. 267–278.

[13] X. Long, M. Kampouridis, and P. Kanellopoulos, "Genetic programming for combining directional changes indicators in international stock markets," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2022, pp. 33–47.

[14] M. Kampouridis and F. E. Otero, "Evolving trading strategies using directional changes," *Expert Systems with Applications*, vol. 73, pp. 145–160, 2017.

[15] O. Salman, T. Melissourgos, and M. Kampouridis, "Optimization of trading strategies using a genetic algorithm under the directional changes paradigm with multiple thresholds," *IEEE XPlore*, 2023.

[16] J. Moody and M. Saffell, "Reinforcement learning for trading," *Advances in Neural Information Processing Systems*, vol. 11, 1998.

[17] K. Lei, B. Zhang, Y. Li, M. Yang, and Y. Shen, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Systems with Applications*, vol. 140, p. 112872, 2020.

[18] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108 014–108 022, 2019.

[19] M. E. Aloud and N. Alkhamees, "Intelligent algorithmic trading strategy using reinforcement learning and directional change," *IEEE Access*, vol. 9, pp. 114 659–114 671, 2021.

[20] N. Alkhamees and M. Aloud, "Dcrl: Approach for pattern recognition in price time series using directional change and reinforcement learning," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021.

[21] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[22] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[23] M. Kampouridis, S.-H. Chen, and E. Tsang, "Market fraction hypothesis: A proposed test," *International Review of Financial Analysis*, vol. 23, pp. 41–54, 2012.