Enhancing High-Frequency Trading with Deep Reinforcement Learning using Advanced Positional Awareness Under a Directional Changes Paradigm

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

***Abstract*—Deep reinforcement learning (DRL) offers the potential to make intelligent trading decisions in high frequency trading strategies in the foreign exchange (FX) market at a fraction of the time it takes humans. In this work, we use an event-based time sampling method referred to as directional changes (DC), which samples data only when there is a significant change in price, to build a DRL-based trading system. The enhanced price representation through DC sampling, combined with positional features reflecting the agent's trading account, provides the DRL agent with information about its exposure to market changes. With this representation of the environment we can train a DRL agent to profitably trade the FX market at high frequencies. Tick data from fourteen FX currency pairs is sampled using the DC framework and then split into windows to form 784 datasets. The novel trading system called PADRL, uses Proximal Policy Optimisation (PPO) to train agents that can autonomously generate considerable levels of profit without rule-based interjections. The resultant agents are compared to four different benchmarks including Buy and Hold, an existing successful DC-based DRL strategy (FDRL) and two popular technical analysis based strategies (MACD and RSI). Strategy performance is measured across three different performance metrics (namely Total Return, Maximum Drawdown and Calmar Ratio), with the novel PADRL system significantly outperforming them all for Total Return and Calmar Ratio.**

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

I. Introduction

The FX market operates 24 hours a day, 5 days a week, and has attracted attention for its financial potential, leading to extensive research on trading strategies. These strategies are often developed through backtesting on historical data, which simulates market conditions to identify profitable methods. Raw prices, representing a currency's value, form the basis for most trading decisions, with each price transaction (or tick) including a timestamp, bid, and ask price. Ticks are sampled to reduce noise, and technical indicators are applied to capture price behaviors. These indicators underpin both rule-based and machine learning (ML) strategies for trading.

Financial indicators and trading strategies have received a lot of attention, while the sampling mechanism that is used has often been overlooked. Fixed interval sampling, often used at regular intervals (e.g., daily, hourly, minute-by-minute), is the most common approach. One such alternative technique is Directional Changes (DC) sampling [1]. This method is motivated by two factors. Firstly, it confirms existing trends by sampling only during significant moves, enabling the construction of rule-based strategies based on this information. Secondly, DC sampling provides clearer information compared to traditional fixed interval sampling by reporting only significant price movements, eliminating many false signals that would be observed using other methods. The DC sampling paradigm relies on a threshold value (θ), expressing a predetermined percentage change in market price considered significant. Key events are recorded as alternating upward and downward directional changes, each subdivided into a directional change (DC) event followed immediately by an overshoot (OS) event [1].

ML can be used to learn the rules that a trader would otherwise have to devise themselves. A key determiner of success in an ML problem is the quality of data. It has been demonstrated that successful trading strategies can be built both with [2] and without [3] the use of DC sampling. Reinforcement learning (RL) is a subset of ML that has been developed to build agents that can take actions in an environment. By simulating the market as an environment and the trader as an agent, RL can be used to learn trading strategies. Deep reinforcement learning (DRL) is a type of reinforcement learning that can use neural networks to learn complex policies [4], [5]. In this work we leverage Proximal Policy Optimisation (PPO), a type of DRL algorithm known for its training efficiency and stability [6], both of which offer significant benefits when dealing with noisy financial data.

In this work we develop a system called PADRL (Position Aware Deep Reinforcement Learning) that uses a collection of DRL agents that learn to trade using DC data and a number of position tracking variables to understand its current exposure to the market and develop a profitable trading strategy in the high frequency FX market. Our aim is to show that the DC sampling algorithm enables the agent to use its inherent ability to learn complex policies, to outperform other benchmark strategies and state of the art strategies. This would therefore demonstrate the value of the combination of both DRL, DC

sampling and advanced positional representation within a high frequency FX environment.

The following sections of this paper are 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 DRL. Section IV describes our methodology. Section V presents the experimental setup. We then present our findings in Section VI. Finally, Section VII concludes the paper and discusses future work.

II. Related Work

The concept of transforming physical time series into DC series was introduced in 1997 by [1]. [1] also empirically formalised 12 DC-based scaling laws using high-frequency data from 13 major FX markets. Subsequent works, such as [7] formalised additional scaling laws. Other works, such as [8] proposed new DC-based indicators, which were used for tasks such as profiling data and identifying regime changes.

An array of ML approaches have been used in conjunction with DC sampling. A large portion of these strategies use evolutionary algorithms such as [9] and [10]. In both works the DC framework outperformed the fixed interval sampling based algorithm. [11] and [12] both use genetic algorithms (GAs) to develop DC based strategies that outperform several non-DC based benchmark strategies. Other ML techniques have also been used to successfully trade a number of different financial instruments [3].

Deep reinforcement learning is a popular technique to apply to problems that can take an agent-based approach, [13], [14]. The finance domain shares many qualities with these agent-based problems, prompting the successful application of DRL in this space [15], [16]. DRL and DC sampling has only been briefly explored in three papers thus far [17]–[19]. Both [17] and [18] use a shallow RL system referred to as 'DCRL', that applies traditional Q-learning techniques 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. [19] uses DRL with DC sampling, however in this work the authors use a trading filter which was necessary to avoid significant losses, as the trading agent was only capable of trading in specific market regimes. Using such a filter poses its own drawbacks as the agent is no longer autonomous and the filter can interrupt the agent's learning of a longer term strategy. In this work we eliminate the need for a trading filter by introducing the notion of positional awareness, providing the agent with a much more information rich representation of state.

III. Background Information

A. Directional Changes

Directional changes (DC) sampling is an event driven data sampling technique for constructing event-based time-series from physical time-series. Traders initially set a threshold value (θ) indicating a significant price change, and successive snapshots of the market are recorded when the price change surpasses this threshold in the opposite direction, forming a time-series that filters out noise between consecutive snapshots. A DC trend includes both uptrends and downtrends, featuring a directional change (DC) event and an overshoot (OS) event. The directional change confirmation (DCC) point marks the tick at which the price exceeds the threshold, differentiating the DC event from the OS event. The end of an OS event is identified retrospectively as a directional change extreme (DCE) point after confirming the next DC event in the opposite direction. The algorithm starts with the initial tick as the DCC point of the first upward movement and then proceeds to iterate through each successive tick, provisionally identifying the new extreme point as the DCE, then finally locking in this value once the DCC in the opposite direction is confirmed. This process then repeats over the remaining data. Figure 1 illustrates the sampling of tick data, represented by the solid grey line, under two different DC thresholds of 0.015% and 0.025%. The smaller threshold of 0.015%, using blue and purple solid lines to represent the DC and OS events respectively, shows how the smaller threshold results in much more frequent sampling of the tick data. The larger threshold of 0.025%, using green and red dotted lines to represent the DC and OS events respectively, shows fewer samples of the tick data as a much larger price move is required to trigger a sample. These two different thresholds offer distinct views on the data and exemplify the adaptability and advantages of DC sampling in financial analysis.

B. Reinforcement Learning

Reinforcement learning (RL) is a machine learning subset that involves an agent interacting with an environment to learn an optimal policy. The agent's policy is the internal decision-making mechanism that generates actions based on received states and rewards. Once a policy has been trained using the states and rewards the observation of the state of the environment can then be passed to the agent and the agent can then use the policy to generate an action. The sequential input and application of actions can be chained together to create a cycle that continues until a termination condition is met. RL algorithms aim to build policies that maximise cumulative rewards over the entire engagement with the environment, also known as an episode.

Traditional RL techniques rely on explicit, rule-based techniques such as Q-learning. Deep reinforcement learning (DRL) combines RL with deep learning, utilising deep neural networks to learn complex policies, ultimately replacing this rule-based mechanism. DRL leverages the universal function approximation capabilities of deep neural networks to handle large, continuous state spaces, something traditional RL techniques struggled with. This makes DRL advantageous for learning policies in environments with expansive and continuous state spaces.

Various different DRL algorithms exist, each with their own benefits and drawbacks for different applications. Proximal Policy Optimisation (PPO) is a specific DRL algorithm de-

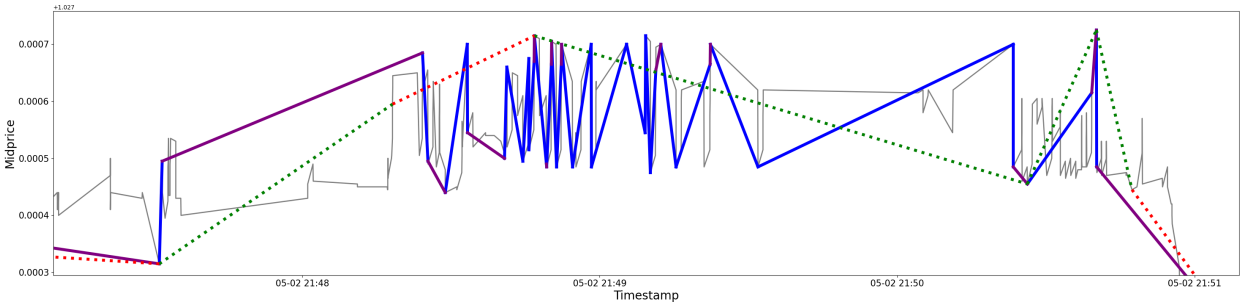

Fig. 1: Directional Changes Sampling Diagram of EUR/CHF at $\theta = 0.015\%$ (solid lines), 0.025% (dotted lines)

signed for more stable learning by taking smaller optimisation steps. PPO employs an actor-critic architecture, where both the actor and critic are represented as deep neural networks. The actor learns the policy (state-to-action mapping), while the critic learns the expected cumulative reward based on states and actions. PPO training involves the actor taking actions, the critic network calculating expected cumulative rewards, and the advantage value being computed to measure the efficacy of the selection actions in a given state. The objective function is then used to apply gradient descent to both the actor and critic networks, improving the policy. For a more detailed algorithm, refer to Algorithm 1.

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

The following methodology is organised as follows. First, in Section IV-A the data preparation phase is discussed. The state representation is then presented in Section IV-B, followed by the definition of the reward function and action space available to the agent in Section IV-C. Finally the performance metrics are discussed in Section IV-D.

A. Data Preparation

The whole set of tick data is first sampled under the DC sampling framework to create a single data set per pair of DC sampled data. This data set then provides the required information to create the DC indicators outlined in Table I. Each indicator-period pair represents a single feature of the data set, this therefore transforms the single dimensional DCC price data time-series to a multi-dimensional time-series data set as we have the DCC price and all the market indicators as the feature set per time step. The feature enriched data is split into contiguous weekly chunks using the physical timestamps. Using this new series of weekly chunks, the combined training, validation and test sets are gathered using a sliding window as shown in Figure 2 (see section V-A for details). Each window sample is offset by the length of the test set to enable the concatenation of the test sets during testing, to report the trading results across the whole duration of the data. Sliding windows are used to enable the models to learn behaviour similar to that of the recent price data.

B. State Representation

The state representation refers to the set of features provided to the agent that represent the state of the agent in an environment at a single point in time. The state representation used in this work consists of a combination of market variables (listed in Table I) and positional variables (described in the next paragraph). The market variables listed in Table I can be calculated prior to training (as described in Section IV-A) to improve efficiency during experimentation. The market variables however can also be calculated in real-time, as this would be required in the application of this high-frequency trading algorithm. Positional variables are dynamic and are based on the prior actions of the agent so must therefore be calculated in real-time during trading. The positional variables are appended to the market variables at the end of each time step to form the feature set from which the agent will make a trading decision for the next time step.

The positional variables consist of the following:

a) Trade Direction: A value representing the direction of the current trade is used to represent if the agent is currently in no position, a long position or a short position, (with the discrete values of 0, 1 and -1 respectively).

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	$\frac{\left(\frac{DCC_t - DCC_{t-1}}{DCC_{t-1}}\right)}{\theta}$	(1, 3, 5, 10)*
$\boldsymbol{R_{DC}}$	Number of ticks adjusted by the return of the event	$\frac{(TMV * \theta)}{\Delta Event_t}$	(1, 3, 5, 10)*
$\boldsymbol{T_{DC}}$	Number of ticks over the course of the event	$\Delta Event_{no.ticks}$	(1, 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)

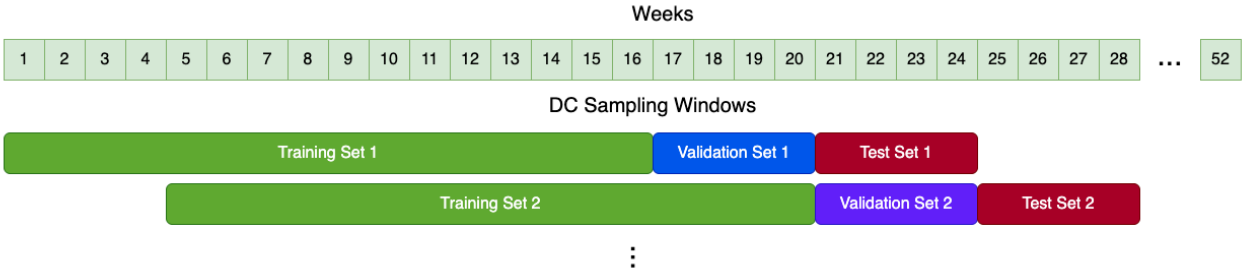

Fig. 2: Diagram of first 2 sampling windows

b) Position Size: The agent is given 100 currency units to begin trading with which is referred to as the agent's balance, the current balance the agent has access to is used to represent the position size the agent is using. The initial balance is subtracted from the current balance to use as the representation of available position size.

c) Potential Return: Potential return is a state space value used to represent the current percentage return the agent would realise given an exit of the current position.

d) Spread: The spread[1] was also used as a positional feature to provide a dynamic measure of market volatility.

It is a requirement of the RL algorithm to bound the feature set between two values. The bounding values in this work are selected as -2 and 2. Bounding the values between two small values like this aids with the convergence and stability of training the neural network inside the agent. The market features are all normalised using z-score normalisation and therefore around 95% of the data fits inside the ± 2 bounds defined in this experiment, resulting in very little information loss for those values that fall outside these bounds, while still maintaining the tight bounds useful for the training of the network. The positional features (current position, balance, potential return and spread) are also bound by the same values. Current position is either -1, 0 or 1 and therefore not affected by the bounds. Spread is re-scaled by a multiplier value specific to the pair in order to fit the bounds. Balance is the variable that loses the most information as the original balance is subtracted to represent profit and is bounded by ± 2 so any information either side of this is compressed into a representation of just -2 or 2. Potential return is often within the same bounds as defined by the problem so there is very little information loss here.

To summarise, after sampling the tick data and adding market features, the data is divided into windows and sets. Real-time positional variables are then incorporated at training time, and all features are constrained within (-2, 2) limits before z-score normalisation. The agent iteratively uses this data, selecting actions with decreasing randomness as training progresses to learn trading strategies for the training set and generalise to the test set. The training is guided by the hyperparameters in Section V-B, and the model is optimised as described in Section V-C.

C. Action and Reward Definition

Action space and reward function are the two other key components of a reinforcement learning system that require careful design. A discrete action space of buy or sell (1, 0) is selected to prevent the agent from holding onto positions from the duration of the episode. This was a behaviour that was observed in preliminary testing that produced inactive systems that would simply not trade as it would immediately result in a negative position due to the transaction cost. If the agent wishes to hold a position, then it will return the action that would result in the position they are currently in (e.g. to hold a buy position a 1 would be returned). The reward function used is the profit realised by the agent when the position is exited as this reward function promotes both active and profitable trading.

D. Performance Metrics

Trading strategies are evaluated using Total Return (see Equation 2) to measure the final return of strategies, Maximum

[1]Spread ($spread = ask\_price - bid\_price$) represents the compensation market makers receive for facilitating trading. Wider spread usually means more trader uncertainty and tighter spread is less trader uncertainty which is a driving force behind market volatility.

Drawdown (see Equation 3) to measure the risk of the strategy by determining how close it came to going bust and Calmar ratio (see Equation 4) to aggregate both total return and maximum drawdown into a risk adjusted return metric[2]. All of these performance metrics are based on the marginal return (see Equation 1) which is the return of each individual trade made during the strategy simulation.

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

where: MR is marginal return, $\%\Delta p$ is the percentage change in price p, 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.

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

where: MDD is maximum drawdown and R is total return. While MR and R are used as metrics to quantify return, and MDD is used to calculate how risky a certain trading strategy is, the Calmar Ratio is used as a measure of risk-adjusted return, as it takes into account both R and MDD. This is a common practice in the literature, rather than considering return and risk in isolation [20].

V. Experimental Setup

A. Data

TrueFX.com[3] is used to download the raw tick data for the fourteen currency pairs presented in the results section. Data for currency pairs including USD are taken from the period 01/01/2022 to 31/12/2022 (AUD/USD, EUR/USD, GBP/USD, NZD/USD, USD/CAD, USD/CHF and USD/JPY). All remaining currency pairs that do not include USD are taken from the period 01/05/2022 to 30/04/2023 (AUD/JPY, CAD/JPY, CHF/JPY, EUR/CHF, EUR/GBP, EUR/JPY and GBP/JPY).

The raw tick data is sampled under eight DC sampling thresholds ranging from 0.015% to 0.029%. This range of DC threshold was selected to provide a large enough number of events while still maintaining a large enough percentage change in price to produce reliable moves. Each sampled series per DC threshold is then split using a sliding window to generate seven windows per threshold per pair, resulting in 784 (14 pairs $\times$ 8 DC thresholds $\times$ 7 windows) windows in total. The training, validation and test sets are generated per window and each window consists of 24 weeks (16 weeks for training, 4 weeks for validation and 4 weeks for testing). Each threshold produces a different sampled data set and testing the trading algorithm on multiple DC thresholds allows us to investigate how effective the trading algorithm is under the DC sampling framework, not just a single DC threshold.

B. Hyperparameter Tuning

Each model is trained for 3,000,000 time steps as an early stopping strategy is employed to determine the best model, so training over a larger number of time steps allows us to identify the best model from larger test space. The 3,000,000 value was determined after a grid search on a smaller data sample size with the intention of finding the number of time steps that provided the best results on the validation set without being too computationally expensive.

The `batch_size` and `n_epochs` hyperparameters were determined using a grid search on a subset of validation sets across a number of pairs. After the grid search it was found that a `batch_size` of 65,536 and a `n_epochs` value of 10 produced the best results without any impractical increase in computational expense.

10% of the total balance is used to enter a position. This value is selected based on a grid search over a reduced set of validation sets to avoid any significant losses during trading as using a larger balance could result in a compounding effect that is significantly detrimental to the performance of the strategy. Future work will investigate a more dynamic position size based on the confidence of the agent.

C. Model Optimisation

There is an inherent level of instability when training using DRL, meaning the performance can go through periods of performance inclines and declines as the algorithm searches through different policy spaces. To account for this behaviour a second validation environment is created and implemented using an evaluation callback[4] in the `Stable Baselines 3` (SB3) reinforcement learning library [21] in Python. The callback is triggered every 50,000 time steps and uses the current model to simulate trading in the validation set. The final balance of the trading simulation over the validation set is recorded. If a validation simulation beats the current highest validation balance of a previous run, then the new model overwrites the previous best performing model. Once training is complete the remaining model is considered the best performing model and is used on the test set.

[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/

[4]A callback is a function allowing us to interrupt the normal training process to conduct our own custom computation.

D. Benchmarks

a) Buy and Hold (B&H): The B&H strategy is a passive strategy often used to provide a useful comparison to strategies that perform active trading. The strategy enters a long position on the first tick and then exits that position on the final tick, making a single trade over the duration of the data.

b) Moving Average Convergence/Divergence (MACD): This strategy relies on the MACD technical indicator [19] by using it to generate buy signals on data sampled at a fixed interval. The fixed interval used for sampling and the indicator period values were identified after a grid search, with a fixed interval of 4 hours. The MACD line, which is defined by the difference of two exponential moving averages, consisted of periods 12 and 26. The signal line, which is defined by the exponential moving average of the MACD line has a period of 9. These chosen parameters produced the highest mean percentage return values across all pairs in the grid search experiment. This benchmark facilitates the comparison of the DC based trading strategies with fixed interval strategies as well as comparing standard technical analysis techniques with DRL techniques for building trading strategies.

c) Relative Strength Index (RSI): This strategy uses the RSI technical indicator [19] calculated at a period of 56, to generate entry and exit signals on the data sampled at a fixed interval of every 30 minutes. The two parameters of indicator period and fixed sampling interval were selected based on the set of parameters that returned the highest mean percentage return after a grid search on appropriate values for each parameter. RSI is a popular indicator used in many technical analysis based strategies and therefore provides a benchmark using fixed interval sampling and technical analysis to compare to DC sampling and RL based strategies.

d) Filtered DRL (FDRL): This strategy trains a DRL agent using the same data that has been sampled under the DC framework. A trading filter is required to trigger the agent when the market exhibits behaviour consisting of periods of short, sharp DC movements in alternating directions. The setup however differs in the novel elements of positional awareness features, the training hyperparameters defined in Section V-B and the lack of trading filter used in PADRL. By using the FDRL system as a benchmark, we are able to compare the performance of a successful DC based DRL system that lacks the novel elements of the PADRL system, therefore demonstrating the benefits of these novel elements.

VI. Results

The following results show the outcome of simulating trading for a years worth of data across the test set of each window, per pair, under a fixed transaction cost of 0.035%.

Table II displays the total return as a percentage observed for the DRL results of PADRL and FDRL strategies and for the fixed interval benchmark strategies for comparison with PADRL. Both DRL strategies from Table II perform favourably on all pairs across all thresholds. The FDRL strategy performs particularly well on pairs that have more periods that are allowed through the trading filter (EUR/CHF, EUR/GBP and EUR/JPY) as these trigger the trading mechanism during periods of short, sharp DC movements. PADRL manages to also perform well on these pairs despite not relying on the rule-based trigger implemented in FDRL. PADRL consistently outperforms FDRL on almost all other pairs across all other thresholds with only a few anomalies. The improved return earned from PADRL can be explained by the profitable trades taken outside the periods that would also trigger trading in FDRL, that have been enabled by the advanced awareness of state that PADRL has over FDRL. When comparing PADRL to the fixed interval benchmark results in Table II, it is clear that the B&H strategy is outperformed on at least one threshold for each pair, and in many cases every threshold, therefore demonstrating that developing the active PADRL strategy is a worthy endeavor over the passive B&H strategy. MACD and RSI also produce relatively low positive returns and some more extreme negative returns across all pairs, suggesting DC with DRL is a more fruitful approach than fixed interval with TA.

The percentage of maximum drawdown (see Equation 3) was used to measure the risk of each trading strategy and the results are displayed in Table III. In this table we can see that RSI is by far the best performing strategy and there is no clear distinction between the two DRL based trading strategies. The RSI strategy is likely showing the best maximum drawdown results due to the conservative nature of the strategy that, although providing very little risk, comes with reduced returns.

The Calmar ratios of all trading strategies are presented in Table IV to demonstrate the risk adjusted return of each DRL strategy. The favourable total return performance of PADRL and the similar maximum drawdown performance of both DRL strategies explains why a similar pattern to the total return results can be observed in Table IV with FDRL performing favourably in the same currency pairs as observed in Table II. We can see that PADRL outperforms all other fixed interval benchmarks in Table IV. Only 3 of the 112 Calmar ratio results presented for PADRL are below the 1.0 level, meaning that total return is greater than maximum drawdown in 97.32% of the pair/threshold pairings.

To assess and compare trading results across various algorithms, we conducted the Friedman non-parametric test for all three performance metrics. This involved calculating the average rank for each algorithm, with lower average ranks indicating better performance. The comparison was made based on performance metric values for each pair-threshold combination across different algorithms, for algorithms without DC thresholds, results were duplicated for the same pair across all DC thresholds. We also carried out the Conover post-hoc test to make more specific comparisons between the algorithms. When we apply Friedman's non-parametric test for total return, as shown in Table Va, we can see that the results are significant due to the reported significance value of 7.89e-55. The Conover post-hoc test presents PADRL as significantly outperforming all benchmark strategies at a significance level of 5%. When observing the maxmium drawdown rankings in Table Vb, there is no clear favourable DRL algorithm as PADRL ranks slightly higher than FDRL as shown by

TABLE II: Total Return (%) by DC threshold (θ) for PADRL (PA) and FDRL (F). B&H, MAC, and RSI strategies are fixed interval based strategies, so only a single value is presented per currency pair. The best value per threshold is denoted in boldface and best value per threshold is underlined.

Pair / θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		B&H	MAC	RSI
	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F			
AUD/JPY	**20.8**	-0.6	**26.3**	2.4	**22.8**	1.5	**24.0**	2.7	**29.8**	3.3	**25.7**	6.9	**36.4**	8.7	**33.6**	7.7	-1.8	-8.5	0.1
AUD/USD	**23.1**	10.3	**56.3**	18.5	**72.5**	26.6	**73.2**	30.4	**28.0**	27.9	**57.7**	35.4	**43.8**	33.6	**59.4**	37.6	-6.3	-11.7	-0.2
CAD/JPY	**22.9**	-1.3	**47.9**	-0.6	**34.7**	-0.6	**27.3**	0.5	**37.1**	0.2	**53.6**	1.8	**49.2**	2.7	**31.8**	3.9	-0.5	-1.4	0.6
CHF/JPY	**8.3**	-0.2	**3.6**	-1.0	**1.1**	0.1	**6.6**	0.1	**9.0**	0.1	**6.4**	-0.7	**9.6**	2.0	**19.8**	3.3	14.0	3.5	-1.2
EUR/CHF	272.6	**340.8**	135.0	**440.9**	127.0	**486.8**	107.5	**916.2**	**1160.8**	652.4	**1222.3**	741.5	**755.4**	612.5	400.1	**1167.6**	-4.0	-3.9	0.2
EUR/GBP	81.6	**158.9**	110.3	**140.8**	167.6	**299.6**	**326.4**	284.4	**615.3**	368.2	**773.8**	394.8	233.0	**467.9**	129.2	**463.4**	4.5	2.2	0.8
EUR/JPY	199.5	**377.7**	385.1	**579.8**	548.5	**801.8**	409.7	**1320.8**	440.3	**1754.3**	331.6	**981.2**	**3291.2**	1918.6	369.4	**1583.9**	9.5	3.3	0.9
EUR/USD	-1.2	**3.8**	**13.0**	10.0	**31.4**	12.2	**20.9**	10.6	**40.6**	12.1	**42.6**	12.1	**43.0**	12.0	**28.8**	14.3	-5.9	4.1	0.6
GBP/JPY	**36.8**	5.4	**37.6**	1.5	**45.8**	8.0	**50.5**	11.5	**53.0**	11.6	**48.8**	16.9	**54.0**	17.0	**29.7**	24.2	4.7	3.0	21.3
GBP/USD	**33.8**	25.0	**47.0**	17.1	**53.8**	27.6	**68.9**	21.5	**66.0**	52.8	**66.1**	46.7	44.4	**54.0**	**61.0**	58.3	-10.6	-3.5	-0.7
NZD/USD	**113.8**	36.6	**70.4**	17.1	**145.7**	46.7	**211.4**	60.9	**278.2**	81.5	53.6	**67.1**	**117.0**	75.4	**226.7**	94.8	-7.1	-16.4	-0.5
USD/CAD	**12.8**	10.3	**13.5**	9.3	10.6	**11.7**	11.7	**12.2**	**23.3**	13.3	**23.1**	11.8	12.8	**13.2**	**27.6**	14.2	7.2	1.6	0.1
USD/CHF	**72.4**	49.8	**133.3**	86.1	**93.7**	90.2	**139.8**	89.6	86.0	**131.5**	127.1	**142.6**	**293.2**	138.5	**268.9**	134.5	1.6	2.2	0.2
USD/JPY	**33.7**	0.5	**13.6**	6.9	**29.4**	7.7	**41.2**	8.1	**25.2**	11.3	**32.5**	15.5	**28.6**	13.5	**30.8**	16.1	13.9	4.4	-2.0

TABLE III: Maximum Drawdown (%) by DC threshold (θ) for PADRL (PA) and FDRL (F). MAC, and RSI strategies are fixed interval based strategies, so only a single value is presented per currency pair. B&H cannot be calculated, as it only performs a single trade. The best value per threshold is denoted in boldface and best value per threshold is underlined.

Pair / θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		MAC	RSI
	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F		
AUD/JPY	1.8	**1.4**	**0.4**	1.0	1.7	**1.5**	3.7	**0.5**	1.4	**0.3**	3.0	**0.6**	1.0	0.7	**0.6**	0.4	13.3	0.2
AUD/USD	**3.3**	9.7	**1.6**	13.5	**4.2**	9.5	**1.6**	8.1	**0.6**	7.8	**1.3**	3.7	**0.8**	6.8	**0.5**	6.1	15.9	0.2
CAD/JPY	7.9	**1.1**	**0.6**	1.1	13.0	**0.7**	11.4	**0.7**	9.2	**1.1**	0.7	**0.6**	**0.6**	**0.6**	17.3	**0.5**	12.4	0.3
CHF/JPY	1.2	**0.7**	5.2	**0.8**	8.5	**0.8**	4.9	**0.6**	4.7	**0.6**	5.7	**1.1**	5.4	**0.4**	1.7	**0.6**	7.3	1.2
EUR/CHF	**3.1**	31.8	**13.6**	13.7	**0.8**	26.5	**0.4**	8.2	7.5	**4.7**	**1.4**	3.4	1.4	**1.3**	**0.7**	0.8	7.0	0.1
EUR/GBP	**15.9**	53.4	**13.6**	33.4	**1.0**	7.2	6.3	**5.1**	**1.4**	4.9	**0.5**	4.2	**1.6**	2.6	**0.4**	1.5	2.7	0.0
EUR/JPY	**1.8**	5.6	**3.1**	3.3	3.3	**1.2**	2.2	**2.4**	11.2	**0.4**	**0.3**	0.6	2.4	**0.3**	2.5	**0.4**	9.7	0.2
EUR/USD	13.5	**4.3**	**5.1**	5.5	5.0	**2.7**	**2.4**	6.7	**0.8**	3.2	**2.3**	4.9	**1.6**	1.8	**1.4**	1.5	6.3	0.1
GBP/JPY	1.1	**0.8**	2.1	**0.6**	3.0	**0.5**	1.0	**0.6**	3.9	**0.5**	3.8	**0.6**	4.0	**0.5**	11.6	**0.3**	5.6	0.0
GBP/USD	23.0	**3.7**	**3.8**	8.4	**2.9**	7.9	17.6	**2.4**	**1.9**	3.0	**0.9**	3.3	**0.8**	2.0	**2.6**	2.7	14.0	2.0
NZD/USD	**1.6**	18.5	**7.5**	9.1	**2.7**	9.1	**4.6**	9.7	**2.2**	8.9	**0.7**	14.0	**0.9**	8.7	**2.1**	10.9	17.9	0.6
USD/CAD	**0.6**	1.9	**0.8**	1.3	**0.6**	1.6	1.1	**0.8**	**0.9**	3.5	**0.4**	2.3	**0.7**	1.7	**0.9**	1.0	3.7	0.0
USD/CHF	**1.5**	20.5	**1.7**	6.3	**1.5**	3.8	**1.9**	8.7	**1.2**	5.6	**1.0**	3.8	**0.6**	1.5	**0.3**	3.8	2.5	0.3
USD/JPY	4.3	**0.8**	3.5	**1.4**	4.1	**0.9**	3.5	**0.5**	3.7	**0.6**	3.5	**0.4**	3.3	**0.4**	3.1	**0.2**	8.9	2.1

TABLE IV: Calmar Ratio by DC threshold (θ) for PADRL (PA) and FDRL (F). MAC, and RSI strategies are fixed interval based strategies, so only a single value is presented per currency pair. B&H cannot be calculated, as it only performs a single trade. The best value per threshold is denoted in boldface and best value per threshold is underlined.

Pair / θ	0.015%		0.017%		0.019%		0.021%		0.023%		0.025%		0.027%		0.029%		MAC	RSI
	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F	PA	F		
AUD/JPY	**11.3**	-0.4	**71.0**	2.5	**13.4**	1.0	**6.6**	5.2	**21.5**	11.5	8.6	**10.9**	**35.0**	12.3	**60.3**	20.6	-0.6	0.5
AUD/USD	**7.1**	1.1	**36.1**	1.4	**17.5**	2.8	**46.3**	3.8	**47.5**	3.6	**46.0**	9.7	**58.0**	4.9	**114.1**	6.2	-0.7	-0.9
CAD/JPY	**2.9**	-1.2	**82.5**	-0.5	**2.7**	-0.8	**2.4**	0.7	**4.0**	0.2	**83.0**	2.8	**82.1**	4.7	1.8	**7.8**	-0.1	1.9
CHF/JPY	**6.9**	-0.3	**0.7**	-1.1	0.1	**0.2**	**1.3**	0.2	**1.9**	0.2	**1.1**	-0.7	1.8	**4.9**	**11.4**	5.3	0.5	-1.0
EUR/CHF	**87.2**	10.7	9.9	**32.1**	**159.5**	18.4	**311.4**	111.2	**155.9**	138.6	**906.9**	219.4	**525.7**	479.9	550.5	**1459.5**	-0.6	1.8
EUR/GBP	**5.1**	3.0	**8.1**	4.2	**162.8**	41.7	51.9	**55.8**	**451.8**	75.4	**1430.6**	94.3	148.3	**179.5**	**372.7**	314.1	0.8	0.0
EUR/JPY	**108.7**	67.9	123.8	**174.9**	164.1	**683.5**	188.6	**553.0**	39.2	**4496.8**	1092.8	**1665.6**	1359.3	**6432.5**	146.7	**3730.6**	0.3	4.9
EUR/USD	-0.1	**0.9**	**2.6**	1.8	**6.2**	4.6	**8.7**	1.6	**50.5**	3.7	**18.3**	2.5	**27.3**	6.8	**21.4**	9.6	0.6	4.9
GBP/JPY	**33.8**	6.7	**18.1**	2.5	15.2	**17.3**	**52.4**	20.0	13.8	**22.1**	12.9	**29.2**	13.6	**37.8**	2.6	**81.8**	0.5	0.2
GBP/USD	1.5	**6.8**	**12.5**	2.0	**18.6**	3.5	3.9	**9.0**	**35.7**	17.6	**72.9**	14.2	**55.1**	27.6	**23.6**	21.8	-0.3	-0.3
NZD/USD	**72.5**	2.0	**9.3**	1.9	**54.9**	5.2	**46.1**	6.2	**125.6**	9.2	**76.7**	4.8	**130.1**	8.6	**109.4**	8.7	-0.9	-0.8
USD/CAD	**19.9**	5.4	**17.9**	7.1	**17.6**	7.1	11.1	**14.4**	**25.6**	3.8	**64.6**	5.1	**18.8**	7.9	**29.8**	14.3	0.4	0.0
USD/CHF	**47.7**	2.4	**77.9**	13.6	**62.0**	23.6	**74.2**	10.3	**69.9**	23.3	**129.1**	38.0	**488.5**	91.5	**780.2**	35.3	0.9	0.7
USD/JPY	**7.8**	0.7	3.9	**5.0**	7.2	**8.9**	11.9	**14.9**	6.8	**19.0**	9.3	**38.5**	8.7	**31.4**	10.1	**73.0**	0.5	-1.0

the rankings of 2.53 and 2.57 respectively. When we apply Friedman's non-parametric test and the Conover post-hoc test to the maximum drawdown values we observe that PADRL and FDRL also do not differ significantly enough to deduce any conclusions from this result. PADRL is however significantly outperformed by RSI as shown by the 1.19 ranking in Table Vb and Conover p-value of $5.43e-32$. When we apply Friedman's non-parametric test and the Conover post-hoc test to the Calmar ratio values (see Table Vc), we observe that the proposed PADRL algorithm ranks first for Calmar ratio and statistically outperforms all other benchmarks as per the p-values of the Conover post-hoc test, as shown by the reported values in bold.

VII. Conclusion

In conclusion, this paper has demonstrated that, with the use of advanced positional awareness features and optimised DRL model development techniques, a significantly profitable and fully autonomous trading system can be built that does

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 post-hoc test. Significantly improved performance over the PADRL strategy at the $\alpha = 0.05$ level is 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 Calmar ratio cannot be defined.

(a) Returns

Friedman test p-value		**7.89e-55**
	Ave. Rank	p_{Con}
PADRL (c)	1.32	-
FDRL	2.15	**6.60e-5**
B&H	3.63	**6.08e-56**
RSI	3.80	**6.04e-62**
MACD	4.10	**2.77e-63**

(b) Maximum Drawdown

Friedman test p-value		**2.15e-46**
	Ave. Rank	p_{Con}
RSI	1.19	5.43e-32
PADRL (c)	2.53	-
FDRL	2.57	9.66e-1
MACD	3.71	**4.25e-23**

(c) Calmar ratio

Friedman test p-value		**1.77e-47**
	Ave. Rank	p_{Con}
PADRL (c)	1.32	-
FDRL	1.92	**6.11e-6**
MACD	3.33	**2.01-67**
RSI	3.43	**1.35e-62**

not rely on rule-based trading filters to make a profit. The results highlight the notable success of the PADRL system, surpassing technical analysis, buy and hold, and DC-based DRL benchmarks in terms of both total return and Calmar ratio. The primary strength of the PADRL system lies in its capacity to yield substantial returns. These high returns, coupled with comparable maximum drawdown performance to the benchmarks, resulted in the PADRL system achieving the most favorable Calmar ratio. This provides evidence supporting the claim that PADRL is the most successful trading algorithm when compared to the benchmarks. While the PADRL system didn't rank first in risk (maximum drawdown), it's crucial to emphasise the importance of a holistic and comprehensive evaluation of trading performance. In this context, the noteworthy point is that PADRL statistically and significantly outperforms all benchmarks when considering the risk-adjusted return metric of the Calmar ratio. This outcome therefore carries greater significance in assessing overall performance.

This approach has produced significantly positive results under the assumptions of fixed transaction costs and no slippage. In real market conditions spread is variable meaning transaction costs may vary and slippage may inhibit the fast execution of orders assumed by these results. This motivates us to build on the approach presented in this paper to move more towards real market conditions. This requires an enhanced level of market awareness by the trading agent that could be obtained by building an ensemble of trading agents that vote on entry and exit points to produce profitable trading algorithms that withstand real FX market conditions.

REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[14] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

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

[16] 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.

[17] 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.

[18] 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.

[19] G. Rayment and M. Kampouridis, "High frequency trading with deep reinforcement learning agents under a directional changes sampling framework," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2023, pp. 387–394.

[20] O. Steinki and Z. Mohammad, "Common metrics for performance evaluation: Overview of popular performance measurement ratios," *Available at SSRN 2662054*, 2015.

[21] 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