Predicting Directional Change Reversal Points with Machine Learning Regression Models

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

Adesola Adegboye
School of Computing
University of Kent
Medway, United Kingdom
adesolaadegboye@gmail.com

***Abstract*—Traditional trading methods often use fixed-interval sampling to capture price changes. In this work, we use an intrinsic time sampling method referred to as directional changes (DC), which reports information whenever there is a significant price change. Tick data from an array of seven FX currency pairs is sampled using the DC framework. We then compare eleven different machine learning (ML) algorithms in a regression task of predicting when the current trend in the market will reverse. These algorithms are: decision tree, random forest, support vector regression, linear regression, stochastic gradient descent regression, kernel ridge regression, elastic net regression, bayesian ridge regression, gradient boosting regression, multilayer perceptron, and long short-term memory neural network. Predicting trend reversal is crucial in trading, as it allows us to anticipate changes in the market and take the relevant actions that are necessary to maximise our returns. After identifying the best ML algorithm for a dataset, we use this prediction as an input of a DC-based trading strategy, and report its performance in terms of return and risk (maximum drawdown). We also benchmark this strategy against four other trading strategies, which include technical analysis and buy and hold. Results over 349 datasets show that the proposed DC-based trading strategy is able to consistently offer high returns at low risk, statistically and significantly outperforming all other benchmarks.**

***Index Terms*—machine learning, regression, directional changes, high-frequency trading**

I. Introduction

The foreign exchange (FX) market is an electronic platform upon which one currency can be exchanged for another. Predicting FX price movement is an important problem in finance that has attracted many researchers due to it's complex nature and opportunity for participants to increase profits at reduced risk. Modern approaches to predicting price movement often involve generating financial indicators from historical data and training a machine learning (ML) model to predict the future price movement. Traditionally, the indicators were derived from tick prices sampled at fixed intervals. Different techniques have been proposed in the literature for creating intrinsic time series such as directional changes (DC) [1], which is the chosen technique in this work. The motivation for using this technique stems from the basis that it has the concept of confirming the existence of a trend whilst the trend is ongoing.

In the directional changes approach, an event summary is generated by recording alternating upward and downward directional changes trends according to a threshold θ. Each trend is subdivided into a directional change (DC) event and an overshoot (OS) event. The DC event is the event that caused a significant price shift either upward or downward. The OS event is an event between two adjacent DC events representing the time between when a DC trend is first confirmed and when a new trend, in the opposite direction begins. The occurrence of a DC trend is confirmed whilst in the trend howbeit in hindsight because its DC event's length is deduced easily whilst the OS event's length is determined after the next DC trend in the opposite direction is confirmed. One of the challenges in the DC approach is the accurate estimation of the OS event length. In this work, we leverage the inherent strength of eleven well-known ML algorithms, and design a multi-ML algorithm trading framework for predicting DC trend reversal.

II. Directional Changes - Background

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.

Figure 1 presents a sample physical time-series converted into DC event series. From the figure, each snapshot, a combination of solid and broken lines is a trend. A DC trend can either be upward or downward. Adjacent blue lines represents downward trends and adjacent red lines represent up trends. The DC trend is composed of a directional change (DC) event (i.e. a solid line) and an overshoot (OS) event (i.e. a broken line). 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.

III. Literature Review

The concept of transforming physical time series into DC event series was first introduced in 1997 by [1], which empirically formalised 12 DC-based scaling laws using high-frequency data from 13 major FX markets. Subsequent works, such as [2] and [3] formalised additional scaling laws. Other works, such as [4], [5] proposed new DC-based indicators, which were used for various tasks, e.g. profiling data, and identifying regime changes.

ML is a popular technical approach for building DC based trading systems, with evolutionary techniques offering the

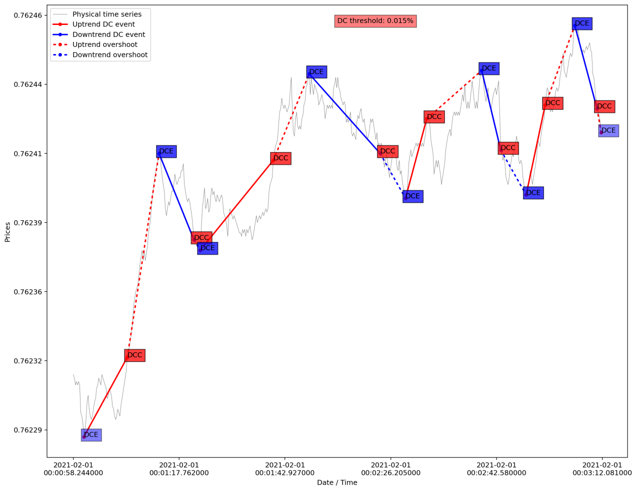

Fig. 1: Directional Change Sampling Diagram of AUDUSD

deepest body of work in conjunction with DC sampling. Works such as [6] and [7], use genetic programming (GP) to build trading strategies. In [6] two GP algorithms were built, one to trade using the DC framework and the other using physical time. The DC framework outperformed the physical time-based algorithm. GP can also be used to solve regression problems, as done in [7] where a genetic programming model is used to conduct a regression task to predict trend reversal points in DC event series.

Genetic algorithms (GAs) have also been used to develop trading strategies in DC event-series [8] and [9]. [8] used GAs to generate trading strategies that outperform technical analysis and Buy and Hold strategies in the FX market. The strategies developed using the DC framework and GAs were shown to outperform the benchmark Buy and Hold and physical time-based technical analysis strategies. [9] used the DC framework to create strategies which were then optimised using GAs. The four DC-based trading strategies introduced in this paper significantly outperformed the benchmark strategies.

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 literature review, it can be deduced that the intersection of the DC event series and evolutionary techniques has been explored to a level of greater depth in comparison to other ML techniques.

IV. Methodology

Our methodology consists of a regression task and a trading task. In the regression task, for a given dataset, we first transform the physical time series (tick data) to DC events using a DC threshold (θ) of 0.015%. We then construct the features and the target variable using the start and end price of the DC move and DC indicators (N_{DC}, C_{DC}, A_T, OSV, T_{DC}, and R_{DC} (see Table I) over various periods as shown in Table II [6]). The end of the OS move is then used as the target variable in the regression task. Once the features and target table have been created, we split the data into training, validation and test sets. We then apply the following eleven ML algorithms: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Linear Regression (LR), Stochastic Gradient Descent Regression (SGD), Kernel Ridge Regression (KR), Elastic Net Regression (EN), Bayesian Ridge Regression (BR), Gradient Boosting Regression (GB), Multilayer Perceptron neural network (MLP) and a Long Short-Term Memory neural network (LSTM). The algorithm that returns the model with the lowest mean absolute percentage error (MAPE) (see Equation 1) in the validation set is then embedded into a DC-based trading strategy.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \tag{1}$$

TABLE I: DC Indicators
Where: θ is DC threshold and DCC is DC confirmation point

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

TABLE II: Feature Set

Indicator	*Period*
TMV	-
OSV	-
$AverageOSV$	(3, 5, 10)
R_{DC}	-
$AverageR_{DC}$	(3, 5, 10)
T_{DC}	-
$AverageT_{DC}$	(3, 5, 10)
N_{DC}	(10, 20, 30, 40, 50)
C_{DC}	(10, 20, 30, 40, 50)
A_T	(10, 20, 30, 40, 50)

In the trading task we define a trading algorithm which we call MLDC (see Algorithm 1). In this strategy we open a position (buy or sell) when the predicted OS end price is beyond the DCC by the threshold θ. We close the position when either the predicted price or the next DCC is reached, whichever occurs first. We also apply a drawdown threshold from the initial balance which will end trading if it is reached to minimise losses. Trading performance is measured per pair for a whole year using total return, maximum drawdown and Calmar ratio. Total return and maximum drawdown give a breakdown of how well the model performs and the risk associated with this performance, the Calmar ratio is then used to represent both these performance metrics in a single value and therefore provide a measure of risk adjusted return. These results are then compared to the trading benchmarks using the same set of performance metrics.

Algorithm 1 MLDC Trading Algorithm

Initialisation: $B, B_{initial} \leftarrow 100$ // Initialise Balance
for DCC **do**
  Obtain OS_{pred} // Predicted price where OS event ends
  if $event$ is $Upturn$ **then**
    if $OS_{pred} \geq DCC + \theta$ **then**
      Enter buy position with whole balance
    end if
  else if $event$ is $Downturn$ **then**
    if $OS_{pred} \leq DCC - \theta$ **then**
      Enter sell position with whole balance
    end if
  end if
  if Next $event$ occurs before OS_{pred} **then**
    Close Position at Next DCC
  else
    Close Position at OS_{pred}
  end if
  if $B \leq 0.001 \times B_{initial}$ **then**
    Stop Trading
  end if
end for

For the regression task the results were compared to a benchmark using the $OS \approx 2DC$ scaling law [1]. For the trading task we compared the MLDC strategy to four different benchmarks AvgOS, B%H, 2DC and TA which include technical analysis, buy and hold and scaling law based trading algorithms (see Section V-B for details).

V. Experimental Setup

A. Data

The tick data for the seven currency pairs is downloaded from TrueFX.com[1] and are taken from the period of 01/02/2021 to 31/01/2022. The seven currency pairs all include USD (AUD/USD, EUR/USD, GBP/USD, NZD/USD, USD/CAD, USD/CHF and USD/JPY). The DC sampling method is applied to the midpoint of the bid and ask prices obtained in the raw data. This sampling method produces a new dataset of contiguous events under the DC framework using a DC threshold (θ) of 0.015% in order to provide enough events to generate reliable results over a large amount data. This fixed threshold is selected after an empirical experimentation process shows this threshold value provides the most information rich dataset.[2]

B. Benchmarks

a) Scaling Law Regression: This regression benchmark uses the ($OS \approx 2DC$) scaling law to develop a regression benchmark that allows us to observe if the ML algorithms are able to produce a more effective regression model than the empirical evidence of such scaling laws.

b) AvgOS: The AvgOS strategy calculates the average number of ticks in the OS events in the training set and then uses this value to predict the end of the OS (and thus trend reversal) on each event in the test set.

c) Buy and Hold (B&H): The B&H strategy enters a long position on the first event and then exits that position on the final event, making a single trade over the duration of the data.

d) 2DC: The 2DC strategy is the trading implementation of the Scaling Law regression method mentioned in Section V-B0a. A position is entered in the direction of a trend at the DCC point. This trade is then exited when the number of ticks since the DCC point is twice that of those during the DC move of the trend.

e) TA: This strategy calculates two MAs, namely a fast and a slow MA, derived from the tick prices for two separate periods. When the fast MA crosses above the slow MA, a buy signal is given and conversely, when it crosses below, a sell signal is given. The periods for the fast and slow MAs used in this work were 70 and 140, respectively [10].

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

[2]Different techniques have been used in the literature to determine an appropriate threshold but in order to test the ability of the novel trading system we have developed, we limited the scope of this paper to a fixed threshold. Future work can be undertaken to develop on this approach with other threshold values.

VI. Results

TABLE III: Mean MAPE (%)

	AUD	EUR	GBP	NZD	CAD	CHF	JPY
Scaling Law	0.018	0.017	0.017	0.020	0.017	0.018	**0.017**
ML Regression	**0.014**	**0.013**	**0.016**	**0.016**	**0.013**	**0.013**	0.019

TABLE IV: Total Return (%). Values in bold face indicate the best return for a given currency.

	AUD	EUR	GBP	NZD	CAD	CHF	JPY
MLDC	**6.71**	**-0.19**	**10.28**	**8.95**	**16.40**	-3.19	4.84
AvgOS	5.82	-1.86	-5.36	3.37	5.32	-4.77	-0.39
B&H	-8.14	-8.94	-2.41	-9.40	-0.48	**2.75**	**8.61**
2DC	-7.31	-3.03	-5.87	-3.74	-4.86	-4.71	-3.59
TA	-0.72	-2.98	1.96	-3.32	0.64	-4.74	-3.43

TABLE V: Maximum Drawdown (%). Values in bold face indicate the best MDD for a given currency.

	AUD	EUR	GBP	NZD	CAD	CHF	JPY
MLDC	4.32	**2.96**	**2.57**	**2.61**	**1.94**	**3.99**	**1.67**
AvgOS	4.73	3.48	5.34	4.77	2.61	5.07	2.69
2DC	7.56	3.16	5.85	4.91	4.94	4.68	3.66
TA	**4.22**	3.02	3.98	4.93	2.53	5.78	3.44

TABLE VI: Calmar Ratio. Values in bold face indicate the best ratio for a given currency.

	AUD	EUR	GBP	NZD	CAD	CHF	JPY
MLDC	**1.55**	**-0.06**	**4.00**	**3.43**	**8.46**	**-0.80**	**2.89**
AvgOS	1.23	-0.53	-1.00	0.71	2.04	-0.94	-0.14
2DC	-0.97	-0.96	-1.00	-0.76	-0.98	-1.01	-0.98
TA	-0.17	-0.99	0.49	-0.67	0.25	-0.82	-1.00

The regression results in Table III show that the ML models perform much better in the regression task than the

TABLE VII: 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}
MLDC (c)	1.285714	-
AvgOS	2.857143	0.084397
TA	**3.142857**	**0.043840**
B&H	**3.428571**	**0.021719**
2DC	**4.285714**	**0.002153**

(b) Maximum Drawdown

Friedman test p-value		**1.55e-19**
	Ave. Rank	p_{Con}
MLDC (c)	1.142857	-
TA	2.571429	0.057584
AvgOS	**2.857143**	**0.025572**
2DC	**3.428571**	**0.004491**

(c) Calmar ratio

Friedman test p-value		**2.53e-36**
	Ave. Rank	p_{Con}
MLDC (c)	1.000000	-
AvgOS	2.285714	0.084559
TA	**3.000000**	**0.010869**
2DC	**3.714286**	**0.001163**

benchmark algorithm in all pairs that were tested, apart from USD/JPY. Due to the fact that each currency pair has around 33-75 rolling windows, we present the mean MAPE of all rolling windows for each currency pair. Table III summarises the mean MAPE results across all currency pairs and rolling windows. The mean MAPE for the Scaling Law Regression across all pairs is 0.0180% compared to the mean MAPE of 0.0149% for the ML predictions. This result is also significant as, when tested over all windows using a Kolmogorov-Smirnov test and a null hypothesis stating that both the Scaling Law benchmark and ML regression approaches produce results from the same continuous distribution, a p-value of 3.253e-137 is returned. This significance value is therefore comfortably within a 0.05 significance threshold and the null hypothesis is rejected.

For the trading task we tested the ML based strategy and four benchmark strategies with a fixed transaction cost of 0.0003% applied to each trade. The return results in Table IV shows that the four benchmarks are outperformed by the ML based trading strategy, where nearly all pairs produce a positive return. The significance of these results can be observed in table VIIa showing that the MLDC algorithm ranks first with an average rank of 1.285714, and the Friedman test returns a p-value of 0.009131, demonstrating that there is statistical significance in the total return ranks. With a Conover post-hoc test (p_{Con}), for $\alpha = 0.05$, the MLDC strategy performs significantly better than all benchmarks apart from the AvgOS strategy, which is significant for $\alpha = 0.1$.

The maximum drawdown Table V shows that the MLDC strategy performs better than the benchmark strategies in all but one pair (AUD/USD). The significance of these results can be seen in Table VIIb where the MLDC strategy can be observed to again rank first with an average rank of 1.142857, and significantly outperforming the 2DC and AvgOS strategies for $\alpha = 0.05$. The TA strategy is also significantly outperformed by the MLDC strategy for $\alpha = 0.1$, as it is marginally over the 5% significance level.

Lastly, the Calmar ratio results in Table VI again show that the MLDC strategy outperforms the four benchmarks across all currency pairs. In Table VIIc, MLDC has an average rank of 1.0, and statistically and significantly outperforms the 2DC and TA strategies for $\alpha = 0.05$. The AvgOS strategy however does not meet the significance level of 0.05 with a p-value of 0.084559, so is significant for $\alpha = 0.1$. As the Calmar ratio is an aggregate measure that considers both risk and return, the consistently high ranking of the MLDC strategy is a very important result, as it demonstrates the algorithm's strength in the risk-return trade off.

Our proposed methodology statistically and significantly outperformed both regression and trading benchmarks. This provides evidence that trading strategies using traditional ML methods to predict the reversal points of DC events can outperform other solely DC-based methods and traditional technical analysis trading methods, as well as the Buy and Hold method.

VII. Conclusion

In conclusion, this paper shows that for results obtained over 349 datasets across 7 FX currency pairs: (i) the proposed MLDC algorithm can lead to profitable trading strategies at low risk, (ii) predicting the trend reversal is a key element to a profitable trading strategy, and (iii) using an array of ML algorithms to predict trend reversal is advantageous, as it allows for the best algorithm to be selected for a given dataset. Further work could include different fixed thresholds or a dynamic threshold system.

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] 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.

[3] M. Aloud and M. Fasli, "The impact of strategies on the stylized facts in the fx market," Citeseer, Tech. Rep., 2013.

[4] E. P. Tsang, R. Tao, and S. Ma, "Profiling financial market dynamics under directional changes," *Quantitative finance, http://www. tandfonline. com/doi/abs/10.1080/14697688.2016*, vol. 1164887, 2015.

[5] 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.

[6] 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.

[7] A. Adegboye and M. Kampouridis, "Machine learning classification and regression models for predicting directional changes trend reversal in fx markets," *Expert Systems with Applications*, vol. 173, p. 114645, 2021.

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

[9] O. Salman, M. Kampouridis, and D. Jarchi, "Trading strategies optimization by genetic algorithm under the directional changes paradigm," in *2022 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2022, pp. 1–8.

[10] 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.