Machine Learning Classification and Regression Models for Predicting Directional Changes Trend Reversal in FX Markets

Adesola Adegboye[a], Michael Kampouridis[b,*]

[a]*School of Computing, University of Kent, Medway, ME4 4AG, UK*
Email: atna3@kent.ac.uk
[b]*Centre for Computational Finance and Economic Agents,*
School of Computer Science and Electronic Engineering,
University of Essex, Wivenhoe Park, CO4 3SQ, UK
Email: mkampo@essex.ac.uk

Abstract

Most forecasting algorithms in financial markets use physical time for studying price movements, making the flow of time discontinuous. The use of physical time scale can make traders oblivious to significant activities in the market, which poses a risk. Directional changes (DC) is an alternative approach that uses event-based time to sample data. In this work, we propose a novel DC-based framework, which uses machine learning algorithms to predict when a trend will reverse. This allows traders to be in a position to take an action before this happens and thus increase their profitability. We combine our approach with a novel DC-based trading strategy and perform an in-depth investigation, by applying it to 10-minute data from 20 foreign exchange markets over a 10-month period. The total number of tested datasets is 1,000, which allows us to argue that our results can be generalised and are widely applicable. We compare our results to ten benchmarks (both DC and non-DC based, such as technical analysis and buy-and-hold). Our findings show that our proposed approach is able to return a significantly higher profit, as well as reduced risk, and statistically outperform the other trading strategies in a number of different performance metrics.

Keywords: Directional changes, Regression, Classification, Genetic programming, Forex/FX, Machine learning

1. Introduction

Financial forecasting is a major activity in financial markets. A big challenge faced by financial traders is the ability to identify security and market trends and so that they can maximise trading returns with minimal associated risk. An enormous amount of research has been dedicated to this topic (Brabazon et al., 2020), but it has been acknowledged from early on that financial time series are among the 'noisiest' and most difficult signals to forecast (Abu-Mostafa and Atiya, 1996). As a result, both the financial (and more recently the machine learning) literature has been continuously looking for new techniques that can lead to better trading results.

*Corresponding author
Email address: mkampo@essex.ac.uk (Michael Kampouridis)

The traditional approach used by financial traders is technical analysis. In this approach, traders use mathematical calculations in identifying and predicting repeating trends in historic data sampled in predetermined physical-time interval (Lin et al., 2017; Samanta et al.). An alternative approach to physical time data sampling is intrinsic time data sampling. In intrinsic series, data is sampled when events considered to be significant occur in the market (Cavalcante et al., 2016; Wan et al., 2016). The idea is that by focusing on important market activities, noise is obfuscated enabling traders build trading strategies around important trends. Over the years, different intrinsic time sampling techniques have emerged, such as perceptual important points (Chen and Chen, 2016; Chung et al., 2001), turning point (Yin et al., 2011), zigzag (Azzini et al., 2010; Özorhan et al., 2018), and directional changes (DC) (Glattfelder et al., 2011; Tsang, 2010; Tsang et al., 2017).

Directional Changes is a relatively new technique, and it has already demonstrated that it can yield profitable returns that can outperform state-of-the-art techniques, such as technical analysis indicators (Kampouridis and Otero, 2017; Aloud, 2016). DC is based on the idea that an event-based system can capture significant points in price movements that the traditional physical time methods ignore. Hence, instead of looking at the market from an interval-based perspective, DC record the key events in the market (e.g., changes in the stock price by a pre-specified percentage) and summarise the data based on these events, moving away from a physical-time view to an event-based-time view. Under this new paradigm, a threshold θ is defined, usually expressed by a percentage of the price. The market is then fragmented and summarised into upward and downward trends. Each of these trends are further dismembered into a directional change (DC) event and an overshoot (OS) event. Different thresholds produce different price summaries. Thus, the directional changes paradigm focuses on the size of price change, while time is the varying factor; whereas in the physical-time paradigm, time was fixed (e.g. daily closing prices).

In a previous work (Adegboye et al., 2017), we used a genetic programming (GP) algorithm to undertake symbolic regression and evolve equations that express linear and non-linear relationships between the length of DC and OS events in a given dataset. The advantage of that approach was that it allowed us to predict when a trend will reverse, and thus increase trading profitability. We used this approach as part of a DC-based trading strategy and tested it over 5 different Forex currency pairs for a total of 250 monthly datasets (5 DC thresholds, over 5 Forex pairs, over 10 months). Our findings showed that our proposed approach was able to outperform other state-of-the-art trading approaches, such as a machine learning algorithm combining a number of technical indicators. This was a major finding, as it was one of the first works to demonstrate the competitiveness of our DC-based trading algorithms against state-of-the-art technical analysis indicators. This has also further motivated us to look for new and better ways to take advantage of the DC framework, as this could lead to improved profitability results.

This work poses an important step forward for more accurate trend reversal prediction, which as we mentioned earlier allows a trader to increase their profitability by being able to anticipate when the current trend will end. The main contribution of this paper is that we do not assume that a DC event is always followed by an OS event, as is often done in the literature. Instead, we create a new step, where we use a classification algorithm to predict whereas a DC event is going to be followed by an OS event. In the end, only when a DC event is classified having a corresponding

OS event, we go ahead with performing symbolic regression. This is an important contribution to our DC framework, as for certain datasets there can be a high number of DC events that are not followed by an OS event. Without this classification step, the symbolic regression GP tends to make conservative estimates of the OS length, as the GP builds equations even for DC events that do not have a corresponding OS event. The addition of the classification step in this work will allow the GP to focus only on DC events that are followed by an OS event, and thus lead to even better end of trend predictions than our previous work (Adegboye et al., 2017).

In addition to the above major contribution, this work also makes the following contributions: (i) We propose a new DC-based trading strategy that uses the combined classification and regression steps, (ii) We do not use the same set of fixed thresholds θ across all datasets. Instead, we use a pool of thresholds and then the best thresholds (in terms of RMSE) are selected for each dataset. Thus, the thresholds we use are tailored to the datasets. (iii) We use a wide range of datasets from 20 Forex currency pairs. In total, our experiments are run over 1,000 different directional changes datasets, making our results much more significant and generalisable. (iv) We add seven new benchmarks and one more performance metric to enhance our results analysis. (v) We present samples of the best equations returned by the symbolic regression and discuss if we can have a generalised equation for predicting trend reversals across different datasets.

The above contributions will allow us to demonstrate not only the effectiveness of DC in terms of generating profitable trading strategies, but also its competitiveness against other state-of-the-art trading techniques. As explained at the beginning of this section, it is important to continuously look for new and improved techniques that lead to more profitable trading results. Our aim is thus to make a novel addition towards the goal of creating more profitable trading algorithms.

To achieve this aim, we have created the following objectives: (i) Demonstrate that the combination of classification and regression leads to error reduction when compared to other trend reversal algorithms, and (ii) Demonstrate that our proposed DC-based trading strategy, which utilises our proposed trend reversal approach, is able to be profitable and outperform other trading strategies, both DC and non-DC-based, including from physical time, such as technical analysis and buy-and-hold. More information about these aims will follow in Sections 3 and 5.

The rest of this paper is organized as follows: Section 2 provides an overview of the DC approach, as well as a discussion on the relevant literature. Section 3 presents all steps of our methodology, namely classification, regression, and trading strategy. Section 4 presents the experimental setup, and Section 5 presents and discusses our findings. Finally, Section 6 concludes the paper and discusses directions for future work.

2. DC Background

2.1. Overview

The directional change (DC) approach is an alternative approach for summarising market price movements. A DC event is identified by a change in the price of a given financial instrument. This change is defined by a threshold

value, which was in advance decided by the trader. Such an event can be either an upturn or a downturn event. After the confirmation of a DC event, an overshoot (OS) event usually follows. This OS event finishes once an opposite DC event takes place. The combination of a downturn event and a downward overshoot event represents a downward trend and, the combination of an upturn event and an upturn overshoot event represents an upturn trend. In other words, a downward trend is a period between a downturn event and the next upturn event and an upturn trend is a period between an upturn event and the next downturn event.

Figure 1 presents an example of how a physical-time price curve is transformed to the so-called *intrinsic time* (Guillaume et al., 1997) and dissected into DC and OS events. As we can observe, two different thresholds are used, and each threshold generates a different event series. Thus, each threshold produces a unique series of events. The idea behind the different thresholds is that each trader might consider different thresholds (price percentage changes) as significant. A smaller threshold captures a higher number of directional changes events, while a higher thresholds captures fewer directional changes events.

Looking at the events generated by a threshold of $\theta = 0.01\%$ (events connected via red lines), we can observe that any price change less than this threshold is not considered a trend. On the other hand, when the price changes above that threshold, then the market is divided accordingly, to uptrends and downtrends. DC events are in solid lines, and OS events are in dashed lines. So a downturn DC event starts at Point A and lasts until Point B, when the downturn OS events starts. The downturn OS lasts until Point C, when there is a reverse in the trend, and an uptrend starts, which lasts until Point D. From Point D to E we are in an upturn OS event, and so on. The end/beginning price point of a new DC trend is called *DC Extreme point (DCE)*; these are Points A, C, E, and $E^{'}$.

As we mentioned, different thresholds generate different event series. Looking at *theta* $= 0.018\%$ (events connected via dotted and dot-dashed lines), we can observe that the events generated are different: a downward trend starts from A and lasts until B′, and the downward OS is from Point B′ until C. Then, from Point C until Point E there is an upward DC trend, and from E to E′ there's an upward OS trend. Algorithm 1 presents the high-level pseudocode for generating directional changes events.

It is important to note here that the confirmation of a change of a trend can only be confirmed retrospectively, i.e. only after the price has changed by the pre-specified DC threshold value θ. For example, under $\theta = 0.01\%$ we can only confirm that we are in a upward trend from Point D onwards. Point D is thus called a *DC Confirmation point (DCC)*. Before Point D, the directional change had not been confirmed (i.e. the market price had not changed by the pre-specified threshold value), thus a trader summarising the data by the DC paradigm would continue believing we are in a downward trend, which started from Point A. Similarly, a trader using $\theta = 0.01\%$ would continue considering being in a upward trend from Point D until the price has reversed by $\theta = 0.01\%$, which only takes place at the next confirmation point, i.e., Point F. So what becomes important here is to be able to anticipate the change of the trend as early as possible, i.e. before Points C and E have been reached. In addition, since different thresholds generate different event series, we hypothesise that the combined information from these series would lead to profitable trading strategies.

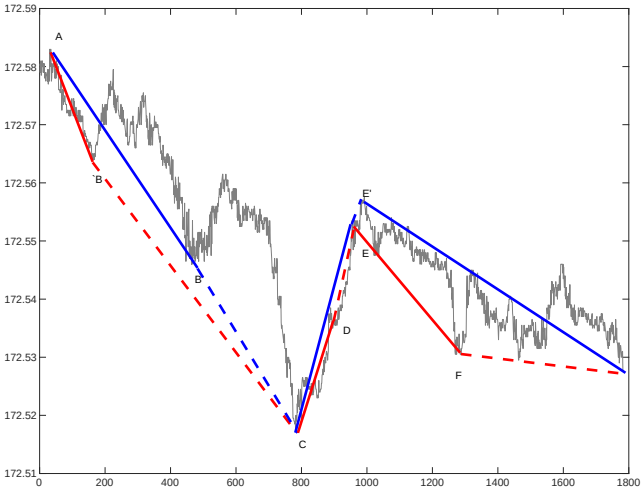


Figure 1: Directional changes for GBP/JPY currency pair. The red lines denote a set of events defined by a threshold $\theta = 0.01\%$, while the blue lines refer to events defined by a threshold $\theta = 0.018\%$. The solid lines indicate the DC events, and the dashed lines indicate the OS events. Under $\theta = 0.01\%$, the data is summarised as follows: Point $A \mapsto B$ (Downward directional change), Point $B \mapsto C$ (Downward overshoot event), Point $C \mapsto D$ (Upward directional change), Point $D \mapsto E$ (Upward overshoot event), Point $E \mapsto F$ (Downward directional change). Under $\theta = 0.018\%$, the data is summarised as follows: Point $A \mapsto B^{'}$ (Downward directional change), Point $B^{'} \mapsto C$ (Downward overshoot event), Point $C \mapsto E$ (Upward directional change), Point $E \mapsto E^{'}$ (Upward overshoot event). Points A, C, E, and $E^{'}$ are DCE points (DC Extreme). Points B, $B^{'}$, D, E, and F are called DCC points (DC Confirmation).

The advantage of this new way of summarising data is that it provides traders with new perspectives to price movements, and allows them to focus on those key points that an important event took place, blurring out other price details which could be considered irrelevant or even noise. Furthermore, DC have enabled researchers to discover new regularities in markets, which are not captured by the interval-based summaries (Guillaume et al., 1997). Therefore, these new regularities give rise to new opportunities for traders, and also open a whole new area for research.

Algorithm 1 Pseudocode for generating directional changes events given threshold Δx_{dc}.

Require: Initialise variables (event is Upturn event, $p^h = p^l = \mathrm{p}(t_0)$, $\Delta x_{dc}(Fixed) \geq 0, t_0^{dc} = t_1^{dc} = t_0^{os} = t_1^{os} = t_0$)

if event is Upturn Event **then**
  if $p(t) \leq p^h \times (1 - \Delta x_{dc})$ **then**
    $event \leftarrow Downturn\ Event$
    $P^l \leftarrow p(t)$ //Price at end time for a Downturn Event
    $t_1^{dc} \leftarrow t$ //End time for a Downturn Event
    $t_0^{os} \leftarrow t + 1$ //Start time for a Downward Overshoot Event
  else
    if $p^h < p(t)$ **then**
      $p^h \leftarrow p(t)$ //Price at start of Downturn event
      $t_0^{dc} \leftarrow t$ //Start time for Downturn event
      $t_1^{os} \leftarrow t - 1$ //End time for a Upturn Overshoot Event
    end if
  end if
else
  if $p(t) \geq p^l \times (1 + \Delta x_{dc})$ **then**
    $event \leftarrow Upturn\ Event$
    $P^h \leftarrow p(t)$ //Price at end time for upturn event
    $t_1^{dc} \leftarrow t$ //End time for a Upturn Event
    $t_0^{os} \leftarrow t + 1$ //Start time for a Upturn Overshoot Event
  else
    if $p^l > p(t)$ **then**
      $p^l \leftarrow p(t)$ //Price at start time for upturn event
      $t_0^{dc} \leftarrow t$ //Start time for a Upturn Event
      $t_1^{os} \leftarrow t - 1$ //End time for a Downturn Overshoot Event
    end if
  end if
end if

2.2. Related DC literature

An example of a regularity that has been discovered by using DC is the relationship between the DC event length and the OS event length (Glattfelder et al., 2011), where the OS length can be expressed as a function of the DC length. This insight can therefore be leveraged by traders for estimating a trend reversal, which is equal to the sum of the DC and OS event lengths, and as a result increase traders' profitability.

More specifically, Glattfelder et al. (2011) observed that the OS event length is on average twice the length of its preceding DC event (Equation 1). Kampouridis and Otero (2017) built on this work by treating the OS and DC length relationships as a linear function with a C constant, where C is the average DC length for a given dataset (Equation 2). Furthermore, because the above two works focused only on linear relationships, Adegboye et al. (2017) created a genetic programming (GP) algorithm to perform symbolic regression and generate both linear and non-linear relationships of DC and OS lengths (Equation 3). The advantage of this approach was that there were no assumptions for the relationship between DC and OS lengths, and it was up to the GP to uncover the function that describes this relationship.

$$OS_l \approx 2 \times DC_l \tag{1}$$

$$OS_l = C \times DC_l; C > 0 \tag{2}$$

$$OS_l = f(DC_l) \tag{3}$$

An interesting observation made in Adegboye et al. (2017), is that it is possible for a DC trend not to have a corresponding OS event. In fact, we have observed that on certain datasets there can be as little as 14.77% of DC trends with a corresponding OS event. This is, of course, threshold-dependent. Nevertheless, the fact remains that one cannot assume that a DC event will always be followed by an OS event; it can be the case that a DC event is followed by another DC event from the opposite direction. This was an important finding because is means that none of the Equations 1-3 are taking this into account; as a result, the symbolic regression GP tends to make conservative estimates of the OS length, as the GP builds equations even for DC events that do not have a corresponding OS event.

Thus, in this work, we introduce a classification task, which is going to predict whereas a DC event will have a corresponding OS event. Only when this is true, we will be applying a GP algorithm to perform the symbolic regression task and derive new formulas, based on Equation 3. The next section presents the classification step, along with the other two steps of our methodology, namely regression and trading strategy.

3. Methodology

As explained in the previous section, there can be a high percentage of DC events that are not followed by an OS event. Therefore, creating a symbolic GP algorithm to predict the length of an OS event as a function f of a DC event (and thus predicting the end of the current trend) over the whole series of DC and OS events is an approach with a major drawback: the resulted function f that describes this relationship does not take into account that many DC events are not followed by an OS event. Hence a function in the form of $\mathrm{OS}_l = 2{\times}\mathrm{DC}_l$ would have been learnt by using inaccurate data.

What we propose in our current work is to clearly separate the cases where a DC event is followed by an OS event (we call this αDC), from those cases that a DC event is followed by another DC event of the opposite direction (e.g. a downwards DC trend is directly followed by an upwards DC trend – we call this βDC). In order to separate these cases, we will be performing classification to predict whether a DC event is followed by an OS event. If the classifier predicts that there will not be an OS event following the DC event, then there is no further action to be taken and the trend reversal point will be the end of the DC event. On the other hand, if DC is followed by OS, then we will use a symbolic regression model to represent the relationship between DC and OS lengths, and thus predict the trend reversal point, which will be at the end of the OS event.

In order to achieve this, there is a number of steps that needs to take place. These steps are summarised in a swim lane diagram in Figure 2. As we can observe, there is a number of operations taking place with the training dataset, and these operations are then feeding into the test (unseen) dataset. More specifically, starting from the physical time dataset (e.g. Forex prices at 10-minute intervals), we apply a GP algorithm, which will return three things: tailored DC thresholds for the specific dataset (A), along with the generated DC dataset (C), and a model (formula) that describes

the relationship between DC and OS length (B). This process related to the symbolic regression GP will be described in detail in Section 3.1. Then, AutoWeka is applied to the DC dataset to decide which is the optimal classification algorithm for the given dataset. Then, the selected model is applied to the test (unseen) dataset, and will predict whether a given DC event is followed by an OS event. Depending on the classification result, the algorithm has two separate prediction points for the end of DC event; in the former case, it predicts that the trend will end at the end of the OS event, whose length was estimated by the symbolic regression GP; in the latter case, since the classifier has predicted that there is not going to be a corresponding OS event, the algorithm considers the end of the DC event as the end of trend, and will thus take a trading action at that point. The above classification process is described in further detail in Section 3.2. Lastly, the final step in our methodology is to use the predicted end of trend point as part of a novel trading strategy, which is presented in Section 3.3.

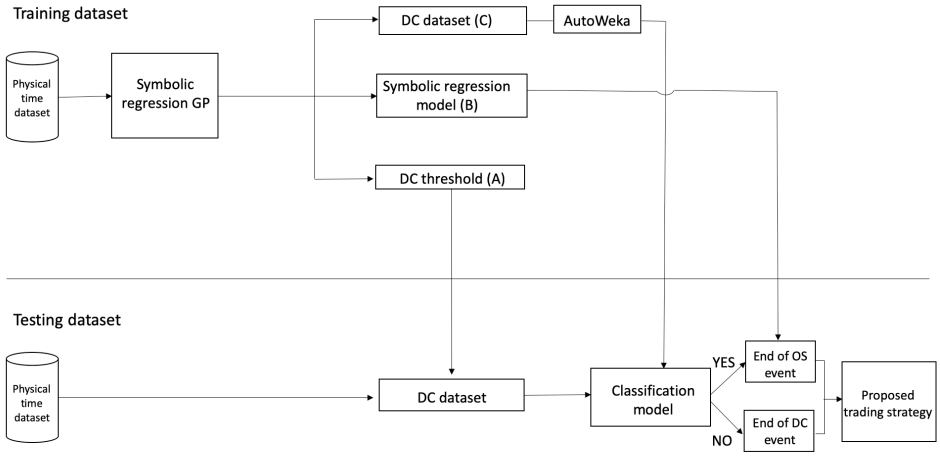


Figure 2: Our proposed methodology.

3.1. GP - Symbolic Regression

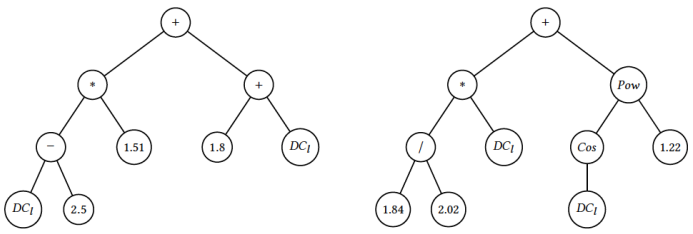
While standard regression techniques, such as linear regression, look for the optimal coefficient values (e.g. a, b) for a given equation form (e.g. $Y = aX + b$), symbolic regression allows us to determine both the functional form and the appropriate coefficients. As a result, we do not need to make any assumptions about the relationship between the given problem variables.

Genetic Programming is considered as state-of-the-art for symbolic regression tasks (Poli et al., 2008). Its ability to build solutions (equations) without any assumptions of their form (i.e. GP is not constrained in building linear equations, or even equations of a specific non-linear form), offers a unique advantage, as it allows it to determine both the functional form of a given dataset, as well as its parameter values. It is based on the Darwinian principle of evolution, where it creates a population of unfit (usually random) programs (equations describing the DC-OS length relationship in our case), and searches the space of mathematical expressions to find linear and non-linear models

that best fit a given dataset. As our DC datasets are bivariate (DC and OS length), we are interested in finding a mathematical equation that best describes the relationship between these two lengths.

Before going into details of the actual GP algorithm, it is important to clarify that even though the GP symbolic regression model is inputed into our framework after the classification step (see Figure 2 above), in terms of implementation, it actually happens first. This is because if we perform the classification task first, then any classification errors are going to affect the effectiveness of the GP models. More specifically, let us assume that we have a dataset of 10 DC events, and that the first 8 DC events are followed by an OS event, whereas the last 2 DC events are not followed by an OS event. Let us also assume that a classifier incorrectly predicts that all 10 DC events are followed by an OS event. In this case, when we apply the GP to perform the symbolic regression task, it will be incorrectly using information (data) from all 10 events in order to construct its models. But what would have been more accurate would be applying the GP only to the DC events that have a corresponding OS event.

For this reason, we perform the GP regression task first under perfect foresight on the training dataset. So we identify those DC events with a corresponding OS event and apply the GP to that data only. The advantage of this approach is that we do not need to deal with the classification task and its corresponding classification errors. Instead, we train the GP only on data that matters, instead of also including noise (i.e. DC events that are not followed with an OS event). Furthermore, the fact that we only do this process on the training dataset avoids having any bias when we eventually apply the selected GP model to the (unseen) test data.



(a) Sample tree 1 **(b) Sample tree 2**

Figure 3: Sample GP individual trees: internal nodes are represented by arithmetic functions. The leaf nodes are represent by numeric constants and the DC length, denoted as DC_l . Given a DC event length the tree estimates the corresponding OS length.

3.1.1. Model representation

We represent our evolved GP individuals using tree structure. Every tree node has an operator function and every terminal (a.k.a leaf) node has an operand, making mathematical expressions easy to evolve and evaluate. We utilise the following 2-arity functions: addition, subtraction, division, multiplication, power and the following 1-arity functions: sine, cosine, power, log, exponential. To prevent the GP from creating invalid solutions (e.g. division by zero), the following functions are protected: division, logarithm, exponential and power. The terminal nodes consist of the DC event length, DC_l, which is given as input, and also an ephemeral random constant (ERC), which is a no-argument

function that returns a random number. All functions and terminals are listed in Table 1.

Figure 3 presents two sample trees from our GP. The first tree represents the equation that calculates OS length as $((DC_l - 2.5) \times 1.51) + (1.8 + DC_l)$ and the second trees represents the equation $((\frac{1.84}{2.02}) \times DC_l) + (\cos(DC_l)^{1.22})$, where DC_l in both equations is the length of DC event. As we can observe, the GP allows us to build both linear (Figure 3 (a)) and non-linear (Figure 3 (b)) relationships between DC and OS lengths. The trees are then going to be evaluated against a given dataset and assigned an error, which indicates how well each tree (model) fits the data.

Table 1: GP Function and Terminal sets

Set	Value
Function set	addition, subtraction, division, multiplication, sine, cosine, power, log and exponential.
Terminal set	DC_l, ERC.
Genetic operation	elitism, subtree mutation and subtree crossover

3.1.2. Model evaluation

It is important to highlight here that the confirmation of a change from upward DC trend to downward DC trend and vice-versa can only be confirmed retrospectively after the price has changed by the pre-specified DC threshold value. Once a directional change is confirmed and a DC trend is classified to be compose of both DC event and OS event, traders are better informed on the potential point in time when DC trend is expected to reverse if OS event length can be adequately estimated. This potential point in time will be the sum of DC event length known at DCC (DC Confirmation point) and OS event length of the αDC estimated using our GP model. To evaluate our GP model we measure the error between actual OS length (OS_l) and estimated OS length ($\hat{OS}_l$). To describe our model performance, we measure the error ε using RMSE shown in Equation 4.

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^{N}(OS_l - \hat{OS}_l)^2}{n}} \tag{4}$$

where n is the sample size.

During evolution, we penalise trees that have only constants as terminal nodes, trees that estimate a negative value and trees that evaluate fitness to NaN or infinity. We perform tournament selection and select parents based on fitness level. We also consider tree depth as a secondary selection criteria in cases where the lowest RMSE is attained by more than one tree. We give preference to the tree with a shorter depth.

3.1.3. Operators and other parameters

We use elitism, subtree mutation and subtree crossover (see Table 1). To control growth, we use hard limits on the depth of offspring programs generated. $Maximum\_depth$ is used for controlling mutation operation.

We have introduced a wrapper to replace incorrect predictions of OS_l with 0. This value was necessary because it was possible for GP to predict negative, NaN or infinite OS length value. We chose the value of 0 because after empirical observations, we realised that there were cases where a DC event is directly followed by another DC event of the opposite direction, hence the OS length of the preceding DC event was 0.

3.1.4. GP outputs

So far we have discussed using the GP to perform the regression task on DC-based data. However, a problem we faced was which DC threshold to use. As we have already mentioned, different DC thresholds produce completely different different DC event series. For example, as we showed in Figure 1, a 0.01% threshold produces different event series to a 0.018% threshold.

In order to decide which thresholds to use during our experiments, we create a pool of DC thresholds. Each threshold then generates a different DC event series. Then, we apply the GP to each one of these event series. As a result we obtain a GP symbolic regression model for each DC event series. Lastly, we rank all GP models in terms of RMSE. This allows us to obtain three outputs: the best GP model, its corresponding DC event series and threshold (i.e. used in generating the corresponding event series). This process is summarised in Figure 4.

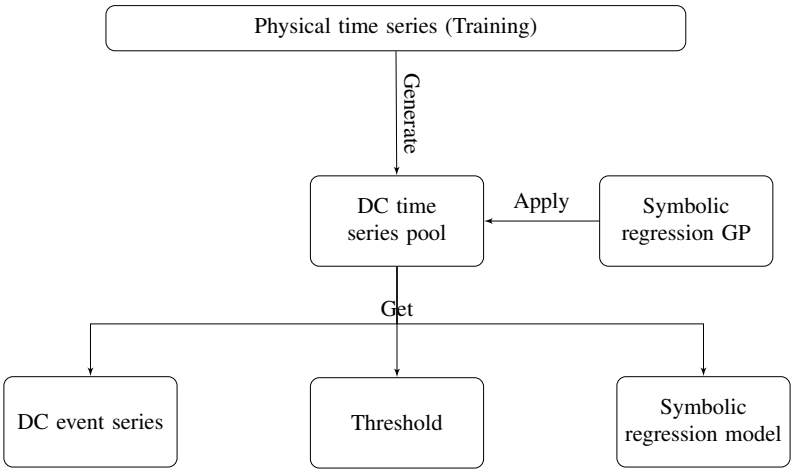


Figure 4: Our proposed framework for evolving symbolic regression model and selecting threshold and DC event with high DC:OS event ratio.

3.2. Classification

The next step in our framework is to classify whereas a DC trend is composed by a DC event and an OS event (αDC), or if a DC trend is solely composed of a DC event (βDC). As there are numerous classification algorithms that can be used for this task, we decided to use Auto-Weka (Thornton et al., 2013), a well-known automated machine learning algorithm. While Weka (Frank et al., 2016), which is a well-known open source package, provides many classification algorithms ready to be used off-the-shelf, there are two important decisions that need to be made: (i) which algorithm to choose for the specific task, and (ii) what are the hyperparameter values for the selected algorithm.

Auto-Weka overcomes this problem, as it uses recent advances in high-dimensional stochastic optimisation to fully automate the above processes. In the end, Auto-Weka returns the best performing algorithm (a classifier in our case), along with an optimal selection of its hyperparameter values with minimal human intervention.

We apply Auto-Weka to each dataset, thus each dataset can end up using a different classification algorithm (Auto-Weka has a choice of 39 classification algorithms). The advantage of this is that we have a tailored classification algorithm and tailored hyperparameters *for each dataset*. To avoid any bias, Auto-Weka is only applied to the training dataset and is trained using 10 k-fold cross-validation. In order to decide which classification model to use for each dataset, we run Auto-Weka 10 independent times per dataset, and in the end we select the classification model with the best f-measure.

The attributes used for the classification task are all DC-related and are presented in Table 2. As we can observe, we use 6 different attributes, which are related to DC and OS events' price and time, as well as the speed of price changing. Attributes X1, X2, were first derived and presented in Glattfelder et al. (2011). Furthermore, attributes X3, X4, X5 and X6 were created for the purposes of this paper, after experimenting with a set of different attributes and identifying the ones with the best classification performance on a training dataset.

Attributes	**Name**	**Description**
X1	DCevent price	This is the price difference between the upturn/downturn point and the directional change confirmation point.
X2	DCevent time	This is the time difference between the upturn/downturn point and the directional change confirmation point.
X3	$Sigma'$	This is the speed at which price change from the start of a trend to directional changes confirmation point.
X4	$DCevent_{-1}$ price	This is the market price at the previous confirmation point.
X5	$DCevent_{-1}$ OS	This is a Boolean variable (Yes/No). Indicates whether the immediate previous DC trend has an OS event.
X6	Flash event	This is a Boolean variable (Yes/No). Indicates whether DC event start time and end time are equa.l

Table 2: Classification attributes - A brief description of independent variables used for classifying whether a DC trend has OS event or not.

The classification process is summarised in Figure 5. As we can see, we use the 'Best DC event series' as input. This is essentially one of the three outputs of the GP process presented in Figure 4 and detailed in Section 3.1. We thus

use this event series to create DC attributes for our classification task. Auto-Weka is then applied to these attributes and at the end we obtain the best classification model for each dataset. A reminder that this whole process takes place for the training data.

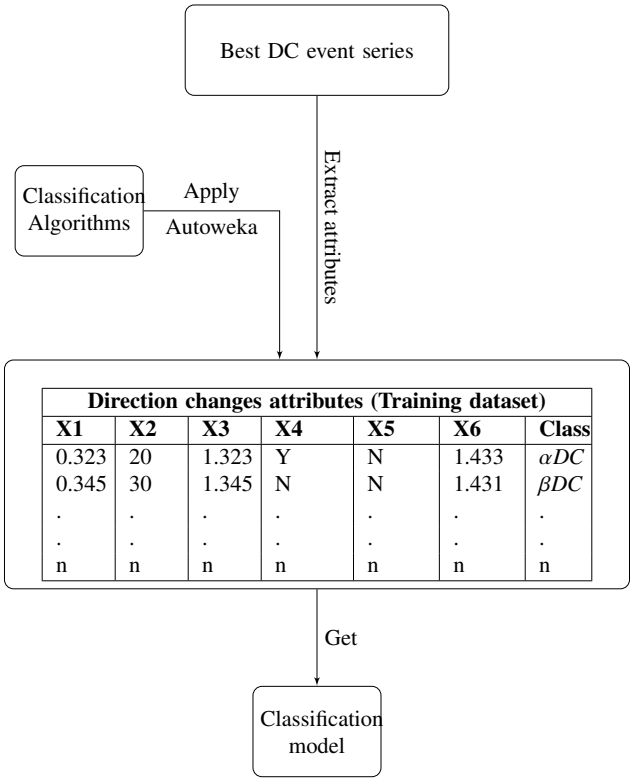


Figure 5: Our proposed framework for creating classification model. The classification model classifies DC trends into αDC and βDC

Once the classification process is also completed, we are then ready to predict the end of a trend in the (unseen) test set by combining the outputs of the classification and the regression steps. What we therefore do is apply the classification model obtained from the training data to the test dataset and classify whether a DC trend is composed by both a DC and OS event (αDC), or not. If the answer is no, then we can predict that the end of the trend will be at the end of the current DC event (DC_l). On the other hand, if a DC event is followed by an OS event, then we can use the symbolic regression model obtained in Section 3.1 and predict the trend reversal point, which is the sum of the DC and OS lengths. This process was also illustrated earlier in Figure 2, which was presented earlier at the beginning of Section 3.

3.3. Trading Strategy

The first two steps presented in the previous two sections allow us to predict the end of a trend in DC event series. To understand how effective this prediction is, we need to use it as part of a trading strategy. In order to do this, we embed our trend reversal prediction process into a trading strategy. For the remainder of this section, we present the trading strategy we use in our experiments.

3.3.1. Trading strategy overview

In order to decide how to trade, we differentiate between opening and closing a position. Opening a position means we sell the base currency and buy the quoted currency. Closing a position means we buy the base currency and sell the quoted currency. We open a position at upward DC trends, provided there is not an existing open position and return is positive after deducting transaction costs. We close a position if there is an existing open position and return is positive after deducting transaction costs. In all other cases we adopt a hold trading strategy. All transactions are done using our entire capital. The transaction cost is 0.025% per transaction. The Opening and Closing strategies are summarised in Algorithms 2 and 3.

Algorithm 2 Trading rules used for selling base currency

Require: Sell rule
  if DC trend is upward **then**
    if There is no open position **then**
      if Is βDC && Return is not negative **then** Open a position at DCC point
      else if Is αDC && DC trend does not reverse before estimated DCE point && Return is not negative **then** Open a position at estimated DCE point
      else Hold
      end if
    end if
  end if

Algorithm 3 Trading rules used for buying base currency

Require: Buy rule
  if DC trend is downward **then**
    if There is an open position **then**
      if Is βDC && Return is not negative **then** Close position at DCC point
      else if Is αDC && DC trend does not reverse before estimated DCE point && Return is not negative **then** Close position at estimated DCE point
      else Hold
      end if
    end if
  end if

3.3.2. Trading strategy evaluation

To evaluate our trading strategy, we measure profitability and risk. We report return, mean maximum drawdown (MDD) and sharp ratio. Return, shown in Equation 5 is the accumulated profit or loss during our trading period and it is calculated by deducting transaction cost from quantity of Forex available for trading and multiplying the result with the exchange rate. Transaction cost shown in Equation 6 is the expense incurred for effecting a trade transaction, in our case the sale or purchase of a foreign currency and it is calculated as 0.025% of the quantity of Forex available for trading. MDD shown in Equation 7 is the downside risk of our strategy and it is measured by calculating the maximum observed loss from a peak price to a trough before a new peak is reached. To measure our excess return, above the

risk-free rate, we use Sharpe ratio shown in Equation 8. It measure the amount of risk involved in obtaining our returns. It is calculated by deduction risk-free rate from mean return and dividing the result by the standard deviation. In this work, we assign 0 as our risk-free rate.

$$R = (Q - TC) \times FXrate \tag{5}$$

$$TC = Q \times \frac{0.025}{100} \tag{6}$$

$$MDD = \frac{P_{trough} - P_{peak}}{P_{peak}} \tag{7}$$

$$SharpeRatio = \frac{R - RFR}{\sigma_R} \tag{8}$$

where R is the return, Q is the quantity, TC the transaction cost, $FRrate$ the FX rate of the relevant currency pair, MDD is the Maximum Drawdown, P_{trough} the trough of the price, P_{peak} the peak of the price, RFR the risk free rate, and σ_R is the standard deviation of the return.

4. Experimental setup

This section is divided into the following parts: Section 4.1 presents the data we are using for the experiments, Section 4.2 presents the tuning configurations for the classification and regression tasks of our framework, and lastly, Section 4.3 presents the setup of the trading experiments.

4.1. Data

We used 10-minute interval high frequency data from March 2016 to February 2017 of the following currency pairs: AUD/JPY (Australian Dollar and Japanese Yen), AUD/NZD (Australian Dollar and New Zealand Dollar), AUD/USD (Australian Dollar and US Dollar), CAD/JPY (CAD Dollar and Japanese Yen), EUR/AUD (Euro and Australian Dollar), EUR/CAD (Euro and Canadian Dollar), EUR/CSK (Euro and Czechoslovak koruna), EUR/NOK (Euro and NOK), GBP/AUD (British Pound and Australian Dollar), NZD/USD (New Zealand Dollar and US Dollar), USD/CAD (US Dollar and Canadian Dollar), USD/NOK (US Dollar and Norwegian Krona), USD/JPY (US Dollar and Japanese Yen), USD/SGD (US Dollar and Singaporean Dollar), USD/ZAR (US Dollar and South African Rand), EUR/GBP (Euro and British Pound). We also used 10-minute interval data from June 2013 to May 2014 of the following currency pairs: EUR/USD (Euro and US dollar), EUR/JPY (Euro and Japanese Yen), GBP/CHF (British Pound and Swiss Franc), and GBP/USD (British Pound and US dollar). All data was purchased from OLSENDATA.com. We considered each month in the period as a separate physical-time dataset. In our tuning phase we used 200 DC

datasets for tuning (i.e. 5 thresholds × 20 currency pairs × first 2 months of our physical-time data). For the rest of our experiment we use 1000 DC dataset (i.e. 5 DC thresholds × 20 currency pairs × remaining 10 months of our physical time datasets). Our tuning and non-tuning DC dataset were split in 70:30 ratio as our training and testing sets.

As different DC thresholds produce different DC event series, we have chosen to evaluate 5 different thresholds for all tuning and non-tuning DC datasets. These thresholds are the best 5 thresholds that are dynamically selected during the GP regression step presented in Section 3.1. In the results section, we will be reporting the average performance of each algorithm, over these 5 DC thresholds.

4.2. Regression and classification algorithms' tuning

The only parameter of Auto-Weka that required tuning was its execution time. This is because Auto-WEKA requires to be given enough time to search its algorithms and hyperparameter space for a classification model that is best in predicting our two class labels (αDC, βDC). We experimented with different runtime configurations namely 15 minutes, 30 minutes, 45 minutes 60 minutes, 75 minutes. We chose a runtime of 60 minutes based on average f-measure, which we observed to diminish at a runtime of 75 minutes. Depending on the number of CPU cores available, it is possible to execute Auto-Weka in serial or parallel mode. For our experiment we executed Auto-Weka in serial mode, using 1 CPU core.

Genetic Programming algorithms have a number of parameters that require tuning, such as the number of the individuals (population), the number of iterations where learning takes place learning (generations), the number of individuals that compete to become a parent in the crossover and mutation operators (tournament size), the probability that we apply genetic operators to a given GP individual (crossover probability, mutation probability, elitism probability), and the maximum depth that a GP tree is allowed to grow (maximum depth). We tuned the above parameters using the I/F-Race package(López-Ibánez et al., 2011), which implements racing methods for the selection of the best configuration for an optimisation algorithm by empirically selecting the most appropriate settings from a set of instances of an optimisation problem(Birattari et al., 2010). Table 3 presents the values of the GP parameters as determined by I/F-Race.

Parameter	
Population	500
Generation	37
Tournament size	3
Crossover probability	0.98
Mutation probability	0.02
Elitism probability	0.10
Maximum depth	3

Table 3: Regression GP experimental parameters for detecting DC-OS relationship, determined using I/F-Race.

4.3. Trading experimental setup

As we have already explained, after completing the classification and regression tasks, our framework can make a prediction on when a DC trend ends. We can then embed this as part of the trading strategy we presented in Section 3.3. This is our proposed algorithm and is called it C+GP+TS.

In order to evaluate the efficiency of our proposed algorithm, we will be comparing it with several other benchmarks. Below we present in detail the different algorithms that we use to benchmark our approach. These benchmarks can be separated into two categories: DC-related algorithms, and non-DC-related algorithms.

4.3.1. DC-related benchmarks

O+TS. This is a DC trend reversal approach originally presented in Glattfelder et al. (2011), where it was observed that on average OS event length is twice the DC event length. In this trading strategy, instead of embedding our classification and regression steps, we embed Equation 1. Thus, the trend reversal point becomes the point where the OS event length is twice the DC length.

M+TS. This is a DC trend reversal approach originally presented in Kampouridis and Otero (2017), where a constant was used to describe the linear relationship between DC and OS length. This constant was tailored to each dataset and separate ratios were calculated for upward trends and downward trends. In this trading strategy, instead of embedding our classification and regression steps, we embed Equation 2. Thus, the trend reversal point is tailored to each dataset.

GP+TS. This is a DC trend reversal approach presented in Adegboye et al. (2017), where symbolic regression GP is used to evolve an equation which represents the ratio between DC event length and OS event length in a dataset. This is the predecessor of our proposed approach, as while it includes the regression step, it does not have the classification step. In this trading strategy, we embed Equation 3.

4.3.2. Non-DC benchmarks

Technical analysis trading strategy. Technical analysis trading strategies is a very popular approach in trading. It uses technical indicators, for insight into when to make trading decisions. We experimented with seven trading strategies that utilise the following indicators; Exponential Movement Average (EMA), Bollinger Bands (BOLLIN), Simple Moving Average (SMA), Aroon Oscillator (AROON), Rate of Change (ROC), Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD), respectively.

Buy and hold. Buy and hold is a well-known benchmark for trading algorithms. Under this trading strategy we bought the quoted currency in the first month of our non-tuning data, then sold it in exchange for the base currency after the 10 month period.

5. Result and analysis

This section presents results for our experiments. It is divided into three main sections: Section 5.1, which presents the classification results, Section 5.2, which presents the regression results, and Section 5.3, which presents the trading results.

We would like to one more time remind the reader that the theoretical contribution of the work are as follows: (i) provide empirical evidence that classification models are effective in detecting the existence of trends before they reverse, thus, providing traders with alternate signal to improve their decision making when trading, (ii) provide empirical evidence that symbolic regression GP models are successful at predicting the point in time when trend is expected to reversal given that the DC trend has an overshoot event, (iii) Demonstrate that by combing our classification and symbolic regression models, prediction error can be considerably reduced when compared to other DC based trend reversal algorithms, (iv) Demonstrate that with our proposed DC-based trading strategy, which utilises our proposed trend reversal approach, profitable trading decisions can be made with reduced risk in comparison to trading strategies experimented . The first and second goals are addressed in Sections 5.1 respectively while the the remainder are addressed in Section 5.3.

5.1. Classification result

We measure the performance of our classification models according to accuracy, precision and recall. Table 4 presents the average results over the 5 DC thresholds and over the 10 months of data, per currency pair. The total average accuracy, precision and recall across the 20 currency pairs we experimented are 0.817, 0.842 and 0.822 respectively. The least average accuracy record currency pair was 0.780 (USD/SGD). The least average precision recorded per currency pair was 0.620 (EUR/CSK), however the remaining 19 average results were all above 0.820. The least average recall recorded per currency pair was 0.419 (EUR/CSK), however the remaining currency pair averages were above 0.713. These results are very important, because they will allow the GP, the next step in our proposed framework, to perform regression only on DC trends that are classified to have a DC and a corresponding OS event (αDC). Therefore, the fact that we have such high values of accuracy, precision and recall for most currency pairs will allow us to obtain better results when predicting the end of a trend. This will become evident in the next section, where the addition of the classification step has led to a much reduced regression error.

5.2. Regression result

Table 5 presents the average RMSE result of the regression step over the 5 DC thresholds and over the 10 months of data, per currency pair. We predict OS event length in DC trends classified as αDC in the classification step. The table also presents currency pair average RMSE results of other OS event length estimation techniques, which we described with Equations 1, 2, and 3. From the table we see that our framework that uses the classification and GP steps (C+GP) consistently outperforms other trend reversal estimators in 13 of the 20 currency pairs. It also ranks second in five cases, behind Equation 3 (the predecessor of our proposed C+GP presented in Adegboye et al. (2017),

Data	Accuracy	Precision	Recall
AUD/JPY	0.851	0.867	0.839
AUD/NZD	0.805	0.827	0.817
AUD/USD	0.829	0.851	0.832
CAD/JPY	0.820	0.825	0.828
EUR/AUD	0.821	0.833	0.866
EUR/CAD	0.839	0.850	0.890
EUR/CSK	0.557	0.620	0.419
EUR/GBP	0.825	0.857	0.857
EUR/JPY	0.821	0.858	0.874
EUR/NOK	0.818	0.841	0.811
EUR/USD	0.806	0.847	0.880
GBP/AUD	0.837	0.896	0.804
GBP/CHF	0.831	0.860	0.861
GBP/USD	0.851	0.858	0.897
NZD/USD	0.848	0.862	0.874
USD/CAD	0.797	0.841	0.829
USD/JPY	0.850	0.863	0.877
USD/NOK	0.887	0.889	0.851
USD/SGD	0.780	0.820	0.816
USD/ZAR	0.877	0.869	0.713
Average	0.817	0.842	0.822

Table 4: Average accuracy, precision and recall results. 1000 datasets consisting of 5 different dynamically generated thresholds tailored to each DC dataset, 20 currency pairs, and 10 months of 10-minute interval data for each currency pair.

which evolved GP symbolic regression models, assuming all DC trends have a corresponding OS event). In addition, C+GP has the lowest average RMSE across all datasets, which is 18.617. This positive result confirms the strength of GP in itself to finding an equation that best represent the relationship between DC and OS event lengths. The introduction of the classification step into the GP step has been proven a successful addition, based on the average RMSE results. To support our findings, we applied Friedman's non-parametric statistical test. The null hypothesis is that the algorithms come from the same continuous distribution. For each algorithm/equation, the table shows the average rank according to the Friedman test (first column), and the adjusted p-value of the statistical test when that equation's average rank is compared to the average rank of the algorithm with the best rank (control algorithm) according to the Hommel post-hoc test (second column). As we can observe, our prosed approach of C+GP ranks first and statistically outperforms at the $\alpha = 0.05$ level Equations 1 and 2. It is also worth noting that Equation 3, which as we have mentioned is the predecessor to our approach, ranks second.

To sup up the findings so far, we can make two important observations: (i) the addition of the classification step

(C+GP) to our existing algorithm that was only using regression to predict the trend reversal has significantly reduced the predictive error, and (ii) our proposed C+GP algorithm ranks first and significantly outperforms two out of the three other trend reversal equations.

Our interest now shifts to using this C+GP as the trend reversal estimation algorithm of a DC-based trading strategy, to investigate whether estimating trend reversal in this manner can lead to an increase trading profit margins.

Algorithms	**C+GP**	**Equation 1**	**Equation 2**	**Equation 3**
AUD/JPY	**15.567**	22.269	25.527	15.627
AUD/NZD	27.368	41.592	51.242	**24.332**
AUD/USD	**11.580**	14.060	16.095	12.814
CAD/JPY	**11.843**	27.251	39.970	18.785
EUR/AUD	21.171	**19.749**	25.728	20.569
EUR/CAD	**16.205**	24.867	23.183	17.719
EUR/CSK	**41.990**	83.845	188.608	52.949
EUR/GBP	24.173	**18.790**	31.430	22.635
EUR/JPY	**19.965**	25.204	28.162	21.117
EUR/NOK	**13.717**	22.499	27.201	13.762
EUR/USD	**28.260**	30.038	38.532	31.061
GBP/AUD	15.138	17.910	21.670	**14.719**
GBP/CHF	**15.961**	23.669	19.358	17.204
GBP/USD	**19.204**	27.778	21.223	24.889
NZD/USD	**10.230**	15.896	14.731	10.588
USD/CAD	26.934	29.315	34.654	**26.818**
USD/JPY	**13.704**	18.326	17.998	14.543
USD/NOK	7.718	10.764	14.128	**7.357**
USD/SGD	**26.932**	34.360	41.712	34.148
USD/ZAR	5.440	7.720	7.796	**4.796**
Average	**18.617**	25.795	34.477	20.265

Table 5: Average RMSE values for each OS length estimator algorithm. 1000 datasets consisting of 5 different dynamically generated thresholds tailored to each DC dataset, 20 currency pairs, and 10 months of 10-minute interval data for each currency pair. Best result per currency pair presented in boldface.

5.3. Trading result

5.3.1. Comparison against DC-based and technical analysis algorithms

Table 7 presents currency pair summary returns of the trading strategies that we detailed in Section 4.3. We would like to draw attention of reader to cases where 0.000s are reported as return in Tables tables 7, 9 and 10; this indicates that for a given currency pair, a hold action was taken by the trading strategy in the 10 months period we experimented.

Algorithm	Average Rank	$Adjust_{pHomm}$
C+GP (c)	1.450	-
Equation 3	**1.850**	0.327
Equation 1	**3.000**	2.932E-4
Equation 2	**3.700**	1.068E-7

Table 6: Statistical test results of OS length estimation according to the non-parametric Friedman test with the Hommel post-hoc test. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

We can see in Table 7 that our C+GP+TS algorithm ranks first in 12 out of 17 currency pairs; in addition, there's 3 pairs (AUD/JPY, CAD/JPY, USD/JPY) that no trading took place. It is also worth noting that with the exception of two currency pairs (EUR/USD, GBP/USD), in all other cases where trading took place, C+GP+TS showed positive returns over the 10-month period of our experiments. It is also worth noting that of the remaining 5 currency pairs, where C+GP+TS did not come first, a DC-based strategy was best in 4. To support our findings, we applied Friedman's non-parametric statistical test. The null hypothesis is that the algorithms come from the same continuous distribution. The result of the statistical test is presented in Table 8 and shows that C+GP+TS is ranking the highest, and that the difference in ranking is statistically significant when it is compared to all other algorithms at the 5% level.

In Table 9 we report the maximum return per currency pair (i.e. 10 months $\times$ 5 thresholds) of the different trading strategies compared. C+GP+TS recorded the highest average maximum return over the 17 currency pairs where trading took place. It had the maximum returns in 10 currency pairs. It can also be observed that all DC based trading strategies record higher maximum return than technical indicator based trading strategies (the only exception is EUR/AUD). In similar manner, Table 10 presents minimum return per currency pair. Results show that C+GP+TS recorded the lowest average negative return (-1.087) amongst DC- based trading strategies over the 17 currency pairs with active trading. However, amongst all trading strategies, Aroon recorded the least average negative return in 11 of the 17 currency pairs where active trading took place. Overall, with regards to mininum returns, the technical analysis indicators have fewer losses compared to the DC-based indicators. It appears that technical analysis indicators take less risks, and thus are able to return lower losses.

Besides trading returns, it is also important to measure the risk involve. To measure risk, we present MDD and Sharpe ratio comparison result which are our metrics for measuring risk. We did not record result for currency pairs AUD/JPY, CAD/JPY and USD/JPY, as active trading did not took place in these markets. Table 11 presents the MDD result. The best average MDD result was recorded by Aaron based trading strategy. The Friedman test presented in Table 12 confirms what we mentioned earlier, i.e. that in terms of risk, technical analysis indicators are more conservative.

To obtain a more holistic view of the results, we also use the Sharpe ratio, which is a well-known aggregate metric of return and risk. Figure 6 presents a chart that details the Sharpe ratio results of the trading strategies. The x-axis presents the time period covered for the relevant currency pair, and the y-axis presents average risk-adjusted return in

percentages. We do not present Sharpe ratio result for markets where there weren't active trading. As we can observe, C+GP+TS is the trading strategy that records the highest number of positive Sharp ratio for a total of 28 times in the 5 month summary chart. Meanwhile, GP+TS, MF+TS, O+TS, EMA, BOLLIN, SMA , AROON, ROC, RSI and MACD had 11, 7, 17, 8, 3, 11, 4, 3, 12 and 4 positive Sharpe ratio results, respectively. Of the 28 positive Sharpe ratio results, 6 where above 0.5, 24 were above 0.2 and the rest were below 0.2.[1] Table 13 presents the Friedman test. It confirms our observations, and C+GP+TS ranks first and statistically outperforms all the other trading strategies at the 5% level.

This is perhaps the most important result so far, as it demonstrates that when considering both returns and risk, our proposed algorithm is able to outperform all other algorithms tested in this paper, both DC-based and also non DC-based. It is also particularly important that C+GP+TS is able to statistically outperform seven well-known technical analysis indicators.

5.3.2. Comparison with Buy-and-hold

Since C+GP+TS has been shown to be the best algorithm across all other DC and technical analysis based algorithms, we will now compare it with the well-known buy-and-hold (BandH) benchmark. The reason we are doing this comparison separately is because for BandH we do not have 10 monthly datasets, as we did with all other algorithms. Instead, we simply buy on the first day of the first month, and sell on the last day of the tenth month.

Table 14 presents comparison trading result between C+GP+TS and BandH. C+GP+TS recorded positive mean annual return in 18 of 20 markets. It outperformed BandH in 12 currency pairs. C+GP+TS's average return across all currency pairs is 0.225% and that of BandH is -0.121%. C+GP+TS reported a variance of 0.153 and BandH's reported a variance of 0.515. This result shows that C+GP+TS is more profitable and less risky that BandH. Finally we performed the Kolmogorov-Smirnov statistical test to investigate whether there is statistical significance in their results. The p-value of the test was 7.2529e-04, which confirms the statistical significance of the performance of C+GP+TS over that of BandH. Thus, the fact that C+GP+TS is more profitable and less risky, outperforming BandH in more markets makes it a more attractive investment strategy.

5.3.3. A sample of best GP models

For completeness, we present some of the equations that our symbolic regression GP evolved below, in their equation format (and not in their tree format). OS_l is the OS length and DC_l is DC length. These four examples are four of our best trees in terms of profitability over all our datasets.

$$OS_l = \log(a + DC_l^b) \quad \text{where a= 1609.55 and b = 5.023.} \tag{9}$$

[1] A ratio of 0.2-0.3 is in line with the general market. A value of 0.5 is considered a market-beating performance if achieved over a long period, a ratio of 1 or better considered superb and difficult to achieve over long periods and a negative Sharpe ratio indicates negative returns.

Trading strategies	C+GP+TS	GP+TS	M+TS	O+TS	EMA	BOLLIN	SMA	AROON	ROC	RSI	MACD
AUD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AUD/NZD	**0.260**	-0.089	-0.063	-0.012	0.002	-0.007	-0.026	-0.002	-0.447	0.056	0.005
AUD/USD	**0.273**	-0.464	-0.327	-0.132	-0.145	-0.393	-0.069	-0.025	-0.321	0.046	-0.147
CAD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EUR/AUD	**0.186**	-0.039	-0.124	0.055	0.057	-0.365	-0.127	0.002	-0.166	-0.060	-0.092
EUR/CAD	**0.192**	-0.243	-0.019	-0.151	-0.226	-0.759	-0.032	-0.068	-0.486	-0.013	-0.346
EUR/CSK	0.034	0.010	**0.035**	0.005	-0.233	-0.067	-0.164	0.000	-0.781	-0.138	-0.281
EUR/GBP	**0.104**	-0.087	-0.061	0.032	-0.135	-0.067	-0.048	-0.061	-0.367	-0.028	-0.240
EUR/JPY	**0.020**	-0.062	-0.020	-0.016	0.015	0.000	-0.027	0.000	0.000	-0.022	0.013
EUR/NOK	**0.351**	-0.043	-0.148	-0.069	-0.118	-0.232	0.149	0.003	-0.261	-0.043	-0.233
EUR/USD	-0.001	0.020	-0.105	**0.032**	-0.492	-0.366	-0.250	-0.064	-0.262	-0.106	-0.409
GBP/AUD	0.354	0.296	-0.247	**0.406**	-0.302	-0.201	-0.022	-0.061	-0.531	-0.159	-0.061
GBP/CHF	**0.202**	-0.116	-0.087	-0.028	-0.268	-0.356	0.009	-0.087	-0.653	0.035	-0.331
GBP/USD	-0.059	-0.048	-0.203	-0.047	-0.076	-0.610	-0.111	-0.045	-0.337	**0.008**	-0.361
NZD/USD	**0.280**	-0.478	-0.159	-0.174	-0.234	-0.445	-0.151	-0.025	-0.333	0.124	-0.366
USD/CAD	**0.044**	0.011	-0.224	0.022	-0.306	-0.640	-0.458	-0.016	-0.708	-0.299	-0.571
USD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
USD/NOK	**0.461**	-0.021	-0.433	-0.019	-0.075	-0.592	0.069	-0.048	-0.700	-0.144	-0.154
USD/SGD	0.030	0.027	-0.023	**0.130**	-0.044	-0.122	-0.178	-0.015	-0.513	-0.057	-0.295
USD/ZAR	**1.762**	0.840	0.695	0.487	0.344	-0.573	-0.356	0.004	-0.057	0.044	0.110
Average Return	0.225	-0.024	-0.076	0.026	-0.112	-0.290	-0.090	-0.025	-0.346	-0.038	-0.188

Table 7: Average return result for trading strategies compared. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best result per currency pair presented in boldface.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+GP+TS (c)	4.408	-
O+TS	**5.338**	3.609E-10
RSI	**5.470**	1.654E-12
AROON	**5.547**	4.805E-14
GP+TS	**5.559**	3.488E-14
M+TS	**5.775**	1.587E-19
SMA	**6.032**	3.884E-27
EMA	**6.196**	1.336E-32
MACD	**6.809**	5.218E-58
BOLLIN	**7.057**	2.329E-70
ROC	**7.806**	3.752E-115

Table 8: Statistical test results of returns according to the non-parametric Friedman test with the Hommel post-hoc test. 10-minute interval out-of-sample date. Significant differences at the $\alpha = 0.05$ level are shown in boldface. BOLLIN is Bollinger bandwidth indicator

Trading strategies	C+GP+TS	GP+TS	M+TS	O+TS	EMA	BOLLIN	SMA	AROON	ROC	RSI	MACD
AUD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AUD/NZD	**2.325**	1.857	1.231	1.387	0.598	0.823	1.242	0.157	0.138	0.347	0.641
AUD/USD	**3.551**	1.481	1.219	1.806	0.324	0.596	0.425	0.000	0.696	0.461	0.445
CAD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EUR/AUD	1.763	1.380	1.089	1.212	0.950	0.107	0.743	0.361	**1.791**	0.200	0.458
EUR/CAD	**2.119**	1.193	2.310	1.374	0.658	-0.117	0.537	0.067	0.059	0.299	0.764
EUR/CSK	0.424	0.449	**0.627**	0.222	0.057	0.000	0.000	0.000	-0.444	0.478	0.000
EUR/GBP	0.906	1.044	1.074	**1.457**	0.541	1.210	0.588	0.000	0.542	0.260	0.554
EUR/JPY	0.433	**0.557**	0.429	0.412	0.154	0.000	0.131	0.000	0.000	0.000	0.135
EUR/NOK	**2.534**	1.216	1.613	0.748	0.081	1.125	1.283	0.074	0.646	0.128	0.853
EUR/USD	**1.004**	0.926	0.785	0.736	0.000	0.206	0.214	0.201	0.476	0.050	0.415
GBP/AUD	2.764	**3.121**	1.247	3.252	0.255	1.136	1.112	0.017	0.262	0.938	1.413
GBP/CHF	**2.065**	1.124	0.940	1.366	0.643	0.174	0.498	0.000	0.041	0.456	0.228
GBP/USD	**1.577**	1.064	0.813	0.925	0.588	0.444	0.313	0.000	0.207	0.302	0.063
NZD/USD	3.059	1.896	**3.683**	1.647	0.489	0.406	0.486	0.000	0.982	0.779	1.017
USD/CAD	1.868	**2.104**	1.614	2.207	1.441	0.420	0.422	0.000	0.368	0.183	0.887
USD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
USD/NOK	**3.273**	1.723	3.054	2.214	0.843	0.652	1.034	0.000	0.026	0.069	0.730
USD/SGD	**1.336**	1.213	0.526	1.678	0.460	0.805	0.000	0.000	0.247	0.330	0.513
USD/ZAR	**7.129**	5.045	4.507	3.601	2.603	0.935	0.743	0.038	2.421	0.684	2.323
Average Maximum	1.907	1.370	1.338	1.312	0.534	0.446	0.489	0.046	0.423	0.298	0.572

Table 9: % Maximum return result for trading strategies compared. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best result per currency pair is shown in boldface. BOLLIN is Bollinger bandwidth indicator

$$OS_l = \log((DC_l \times a)^b) \quad \text{where a= 4.117 and b = 5.764.} \tag{10}$$

$$OS_l = \cos(a \times \cos(DC_l)) + \frac{b}{exp(\cos(DC_l)} \quad \text{where a = 292.160 and b= 4.569} \tag{11}$$

$$OS_l = \exp(\exp(\sin(\sin(DC_l)))) + (a \times (b + \log(DC_l))) \quad \text{where a = 1.750 and b = 1.957.} \tag{12}$$

Trading strategies	C+GP+TS	GP+TS	M+TS	O+TS	EMA	BOLLIN	SMA	AROON	ROC	RSI	MACD
AUD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
AUD/NZD	-1.189	-1.559	-1.048	-1.578	-0.482	-0.620	-0.507	**-0.120**	-0.836	-0.170	-1.040
AUD/USD	-1.918	-3.646	-3.240	-1.143	-1.022	-1.375	-0.578	**-0.136**	-1.265	-1.037	-1.170
CAD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
EUR/AUD	-0.632	-2.024	-1.092	-2.249	-0.605	-1.059	-0.695	**-0.209**	-1.013	-0.560	-0.722
EUR/CAD	-1.299	-1.801	-1.446	-1.727	-1.285	-1.717	-0.741	-0.477	-1.123	**-0.298**	-2.068
EUR/CSK	-0.168	-0.202	-0.197	-0.037	-0.670	-0.200	-0.315	**0.000**	-1.586	-0.380	-0.626
EUR/GBP	-0.619	-1.206	-0.743	-0.970	-0.503	-1.245	-0.648	-0.454	-0.855	**-0.376**	-0.992
EUR/JPY	0.000	-1.344	-0.658	-1.185	**0.000**	0.000	-0.397	0.000	0.000	-0.222	**0.000**
EUR/NOK	-0.863	-1.490	-1.743	-1.404	-0.499	-1.282	-0.650	-0.043	-1.177	**-0.411**	-1.182
EUR/USD	-1.189	-0.744	-1.032	-0.563	-1.181	-1.032	-1.229	**-0.326**	-0.882	-0.940	-1.102
GBP/AUD	-1.506	-2.736	-2.368	-1.800	-1.250	-1.610	-0.837	**-0.361**	-2.061	-1.928	-0.755
GBP/CHF	-0.811	-1.542	-1.274	-0.969	-1.546	-1.072	-0.469	-0.359	-2.566	**-0.101**	-0.994
GBP/USD	-0.938	-2.396	-1.009	-2.665	-0.658	-1.293	-0.588	-0.390	-1.052	**-0.200**	-0.774
NZD/USD	-1.758	-2.556	-1.631	-2.612	-1.659	-1.952	-0.995	-0.086	-1.265	**0.000**	-1.028
USD/CAD	-1.870	-2.625	-3.111	-2.758	-2.010	-1.974	-2.351	**-0.156**	-1.698	-2.092	-2.501
USD/JPY	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
USD/NOK	-3.141	-1.644	-5.282	-1.374	-1.136	-2.051	-0.851	**-0.343**	-1.534	-0.935	-1.669
USD/SGD	-0.724	-0.647	-0.847	-0.493	-0.567	-0.861	-0.528	**-0.146**	-1.462	-0.544	-0.762
USD/ZAR	-3.105	-4.979	-3.339	-5.292	-2.164	-2.201	-1.204	**0.000**	-3.504	-0.702	-2.679
Average Minimum	-1.087	-1.657	-1.503	-1.441	-0.862	-1.077	-0.679	-0.180	-1.194	-0.545	-1.003

Table 10: % Minimum return result for trading strategies compared. 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best result per currency pair shown in boldface. BOLLIN is Bollinger bandwidth indicator

As we can observe, most equations have different structures. The first two are logarithmic equations, whereas the third has both the cosine and the exponential functions, and the fourth equation has exponential, sine and logarithmic functions. This is particularly interesting and important, because it demonstrates that the relationship between the DC and OS lengths can be non-linear and also dependent on the dataset. Thus our work of using classification and regression to predict the OS length has allowed us to uncover this relationship for each dataset and increase the profitability of the trading strategies.

5.3.4. Computational times

Table 15 presents the average computational times for all algorithms. We should note that because MF+TS, O+TS, EMA, BOLLIN, SMA , AROON, ROC, RSI and MACD and BandH are deterministic algorithms and don't have the extra step of training models, their execution is faster taking from around 3 seconds to 30 seconds. C+GP+TS and GP+TS are non-deterministic algorithms, thus their execution times vary between around 5 and 70 minutes. The

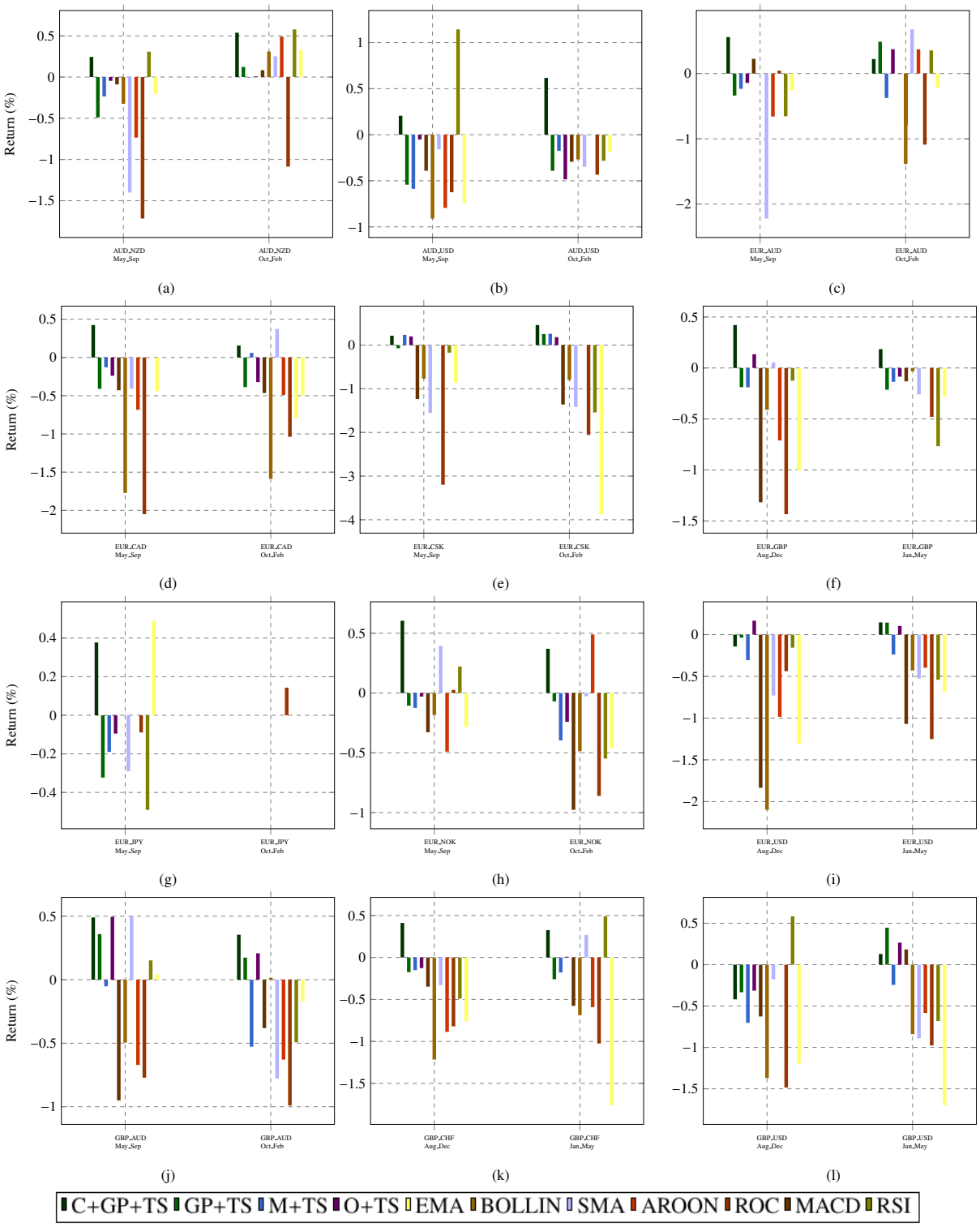
Trading strategies	C+GP+TS	GP+TS	M+TS	O+TS	EMA	BOLLIN	SMA	AROON	ROC	RSI	MACD
AUD/NZD	0.123	0.151	0.159	0.206	0.170	0.368	0.286	**0.018**	0.681	0.071	0.159
AUD/USD	0.160	0.312	0.148	0.275	0.118	0.634	0.228	**0.025**	0.622	0.162	0.170
EUR/AUD	0.106	0.155	0.077	0.206	0.120	0.496	0.380	**0.041**	0.508	0.104	0.173
EUR/CAD	0.135	0.258	0.203	0.308	0.123	0.823	0.172	0.089	**0.748**	0.080	0.148
EUR/CSK	0.006	0.008	0.013	0.002	0.036	0.074	0.166	**0.003**	0.783	0.186	0.039
EUR/GBP	0.100	0.078	0.137	0.086	0.242	0.372	0.204	**0.064**	0.532	0.072	0.196
EUR/JPY	0.011	0.038	0.008	0.052	0.004	0.000	0.040	0.000	0.000	0.022	0.001
EUR/NOK	0.133	0.148	0.256	0.152	0.114	0.469	0.186	**0.004**	0.688	0.090	0.101
EUR/USD	0.155	**0.069**	0.126	0.083	0.224	0.495	0.299	0.095	0.416	0.191	0.223
GBP/AUD	0.191	0.239	0.201	0.177	0.208	0.731	0.273	**0.081**	1.030	0.283	0.216
GBP/CHF	0.096	0.106	0.049	0.191	0.163	0.525	0.281	0.087	0.780	**0.016**	0.200
GBP/USD	0.132	0.180	0.122	0.162	0.122	0.842	0.340	**0.077**	0.508	0.112	0.270
NZD/USD	0.289	0.324	0.266	0.404	0.270	0.861	0.385	**0.026**	0.698	0.000	0.263
USD/CAD	0.168	0.162	0.340	0.239	0.323	0.898	0.609	**0.024**	0.906	0.333	0.167
USD/NOK	0.141	0.175	0.569	0.172	0.235	1.051	0.245	**0.054**	0.928	0.173	0.099
USD/SGD	0.077	0.074	0.069	0.036	0.127	0.316	0.267	**0.015**	0.590	0.152	0.079
USD/ZAR	0.117	0.145	0.281	0.331	0.357	1.476	0.628	**0.000**	1.069	0.149	0.709
Average MDD	0.107	0.131	0.151	0.154	0.148	0.522	0.249	**0.035**	0.574	0.110	0.161

Table 11: %Average maximum drawdown results for 10-minute interval out-of-sample data. 20 different currency pairs and 10 calendar months each representing the physical dataset. 5 DC dataset were generated using 5 dynamically generated thresholds tailored to each DC dataset. Best result per currency pair shown in boldface.

Trading strategies	Average Rank	$Adjust_{pHomm}$
BOLLIN (c)	2.162	-
AROON	**3.291**	2.635E-14
RSI	**5.639**	2.952E-121
C+GP+TS	**5.962**	2.769E144
O+TS	**5.338**	1.693E-146
GP+TS	**6.171**	3.106E-160
M+TS	**6.181**	5.994E-161
EMA	**6.771**	3.802E-211
MACD	**7.099**	4.514E-242
SMA	**7.342**	2.611E-266
ROC	**9.386**	0.0

Table 12: : Statistical test results of maximum drawdown of DC based trading strategies according to the non-parametric Friedman test with the Hommel post-hoc test. 10-minute interval out-of-sample data. Significant differences at the $\alpha = 0.05$ level are shown in boldface.

higher computation times for C+GP+TS was expected. This is because we configured AutoWeka to run for 60 minutes in order to search for the most suitable classifier and optimise its hyperparameters which we also mentioned in Section


Return (%)
AUD_NZD May_Sep
AUD_NZD Oct_Feb
(a)
AUD_USD May_Sep
AUD_USD Oct_Feb
(b)
EUR_AUD May_Sep
EUR_AUD Oct_Feb
(c)
EUR_CAD May_Sep
EUR_CAD Oct_Feb
(d)
EUR_CSK May_Sep
EUR_CSK Oct_Feb
(e)
EUR_GBP Aug_Dec
EUR_GBP Jan_May
(f)
EUR_JPY May_Sep
EUR_JPY Oct_Feb
(g)
EUR_NOK May_Sep
EUR_NOK Oct_Feb
(h)
EUR_USD Aug_Dec
EUR_USD Jan_May
(i)
GBP_AUD May_Sep
GBP_AUD Oct_Feb
(j)
GBP_CHF Aug_Dec
GBP_CHF Jan_May
(k)
GBP_USD Aug_Dec
GBP_USD Jan_May
(l)
C+GP+TS
GP+TS
M+TS
O+TS
EMA
BOLLIN
SMA
AROON
ROC
MACD
RSI

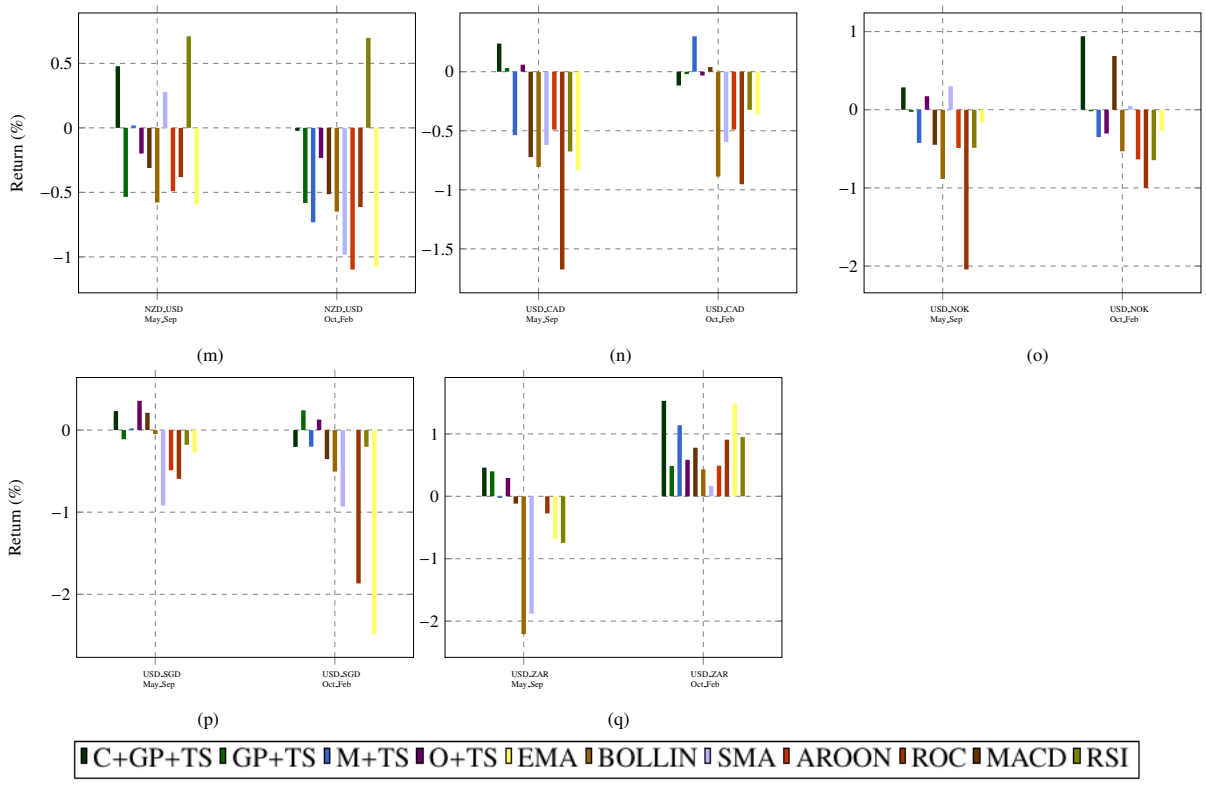


Figure 6: Average Sharpe ratio for all currency pairs.

Trading strategies	Average Rank	$Adjust_{pHomm}$
C+GP+TS (c)	2.399	-
O+TS	**3.971**	0.047
GP+TS	**5.000**	0.002
M+TS	**5.314**	7.112E-4
RSI	**5.600**	2.173E-4
SMA	**6.229**	6.860E-6
EMA	**6.357**	3.600E6
AROON	**6.600**	8.217E-7
MACD	**7.529**	7.906E-10
BOLLIN	**8.000**	1.462E-11
ROC	**9.000**	8.457E-16

Table 13: Statistical test results of the Sharpe ratio results according to the non-parametric Friedman test with the Hommel post-hoc test. 10-minute interval out-of-sample date. Significant differences at the $\alpha = 0.05$ level

4.3.[2]. Auto-WEKA can be executed in both single-threaded and multi-threaded modes. We have chosen to perform our experiments using single threaded mode due to limited hardware resources. The minimum execution time for

[2]The time taken in the classification phase of C+GP+TS went above the allotted time of 60 minutes due to CPU time slice as other processes were running on the hardware simultaneously.

Trading strategies	C+GP+TS	Buy-and-hold
AUD/JPY	**0.000**	-6.278
AUD/NZD	**0.260**	-0.516
AUD/USD	**0.273**	-5.728
CAD/JPY	**0.000**	-4.109
EUR/AUD	**0.186**	-2.672
EUR/CAD	0.192	**18.555**
EUR/CSK	0.034	**7.770**
EUR/GBP	**0.104**	-0.292
EUR/JPY	**0.020**	-6.211
EUR/NOK	0.351	**2.046**
EUR/USD	-0.001	**8.801**
GBP/AUD	0.354	**3.936**
GBP/CHF	**0.202**	-2.395
GBP/USD	-0.059	**8.464**
NZD/USD	**0.280**	-6.443
USD/CAD	0.044	**2.345**
USD/JPY	**0.000**	-9.430
USD/NOK	**0.461**	-6.102
USD/SGD	0.030	**0.207**
USD/ZAR	**1.762**	-4.505
Mean	**0.225**	-0.128

Table 14: % Mean trading result of C+GP+TS vs Buy-and-hold trading strategies per currency pair. 10-minute interval out-of-sample data. Results show RMSE value. They are averaged over 5 different dynamically generated thresholds tailored to each DC dataset and 20 currency pairs.

Trading strategies	C+GP+TS	GP+TS	M+TS	O+TS	EMA	BOLLIN	SMA	AROON	ROC	RSI	MACD
Classification	~ 65 mins	–	–	–	–	–	–	–	–	–	–
Regression	~ 5.45 mins	~ 6.2 mins	~ 30 secs	~ 20 secs	–	–	–	–	–	–	–
Trading	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec	~ 3 sec

Table 15: Average computational times per run for C+GP+TS, GP+TS , M+TS , O+TS, EMA, BOLLIN, SMA, AROON,ROC,RSI and MACD. BH is deterministic algorithms and only take around 1 second to execute.

Auto-WEKA is 1 minute. With the right amount of hardware resources and executing Auto-WEKA in multithreaded mode, this huge amount of time spent on searching for suitable classifier can be reduced significantly. The regression task of C+GP+TS took only 5.45 minutes on average, which is comparable to its predecessor's 6.2 minutes. Since GP is involved, it is not surprising that additional time is needed to evolve a suitable symbolic regression model. MF+TS and OS+TS on the other hand make is a simple calculation once the DC event length is determined, hence the speed in regression phase. To further improve computation time taken by the evolutionary algorithm (GP), we can also

parallelise the training which can reduce the time taken by up to 21 folds Brookhouse et al. (2014). Lastly, the trading task has the same duration across all algorithms: less than 3 seconds.

It is also important to note that for trading we would normally perform the learning processes on the training data off-line, then apply the best model to the test data. Thus, the fact that classification and regression last around 70 minutes is not a problem, as for the actual trading we would simply be applying the trading model, and thus its execution would be on par with all other algorithms' execution time, including the technical analysis indicators. Finally given the significant improvements that we have observed in terms of increase in returns and reduction in risk, this slower execution time in this experimentaion scenario is justified.

5.4. Summary

Based on our experimental results, we can reach the following conclusions.

Using classification algorithms and GP for symbolic regression is an effective way of predicting the trend reversal in DC summaries. As we observed in Tables 4 - 6, the very positive classification results have led to significant reduction in RMSE, ranking our proposed C+GP first against all other DC-based trend reversal algorithms.

Utilising the above trend reversal algorithm as part of a trading strategy can lead to profitable results. In fact, C+GP+TS significantly outperformed all other trading strategies in a variety of metrics, such as mean and maximum returns.

Our proposed trading strategy is one of the least risky strategies. As we saw from both MDD and Sharpe ratio results, C+GP+TS is a risk aversive strategy even though it prioritised market profit over capital preservation and this is confirmed in Sharpe ratio results where it outperformed other algorithms. The above thus lead us to conclude that C+GP+TS is a trading strategy with very low risk when compared to all other strategies presented in this work.

There is no generalised formula for predicting trend reversal in DC-based summaries. This is perhaps the most important finding of our work. This is because it demonstrates that each dataset can have its own unique characteristics, and predicting trend reversal requires tailored solutions and not equations that are applied to all trends, irrespectively of their characteristics.

6. Conclusion

To conclude, this paper presented a new framework, where we used different machine learning algorithms for classification and regression in DC-based summaries, to predict end of trend. This then enabled us to develop profitable and low-risk trading strategies, which were able to outperform six benchmarks, including other DC-based trading strategies, technical analysis, and buy and hold. It is important to note here that we run extensive experiments over a total of 1,000 datasets from 20 different Forex currency pairs. This thus leads us to believe that our results are not only significant, but also widely applicable.

Future work will focus on combining multiple DC thresholds under a single trading strategy. As we have explained, each DC threshold creates a different summary. In our current work, we experimented with 5 different

thresholds and presented their average results. We believe that it would be interesting to combine the 'knowledge' of multiple thresholds under an optimisation algorithm, and investigate how/if the trading strategies' profitability can further increase. Other research directions could also be to create a tailored classification algorithm (instead of using Auto-Weka's out-of-the-box algorithms) and further improve our GP algorithm, to reduce the error of predicting end of trend even more, and thus lead to more profitable trading strategies.

Project code: The source code for this project can be found in a GitHub repository, at the following address:

`https://github.com/adesolaadegboye/SymbolicRegression`

References

References

Abu-Mostafa, Y.S., Atiya, A.F., 1996. Introduction to financial forecasting. Applied intelligence 6, 205–213.

Adegboye, A., Kampouridis, M., Johnson, C.G., 2017. Regression genetic programming for estimating trend end in foreign exchange market, in: Computational Intelligence (SSCI), 2017 IEEE Symposium Series on, IEEE. pp. 1–8.

Aloud, M.E., 2016. Profitability of directional change based trading strategies: The case of Saudi stock market. International Journal of Economics and Financial Issues 6.

Azzini, A., da Costa Pereira, C., Tettamanzi, A.G., 2010. Modeling turning points in financial markets with soft computing techniques, in: Natural Computing in Computational Finance. Springer, pp. 147–167.

Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T., 2010. F-race and iterated f-race: An overview, in: Experimental methods for the analysis of optimization algorithms. Springer, pp. 311–336.

Brabazon, A., Kampouridis, M., O'Neill, M., 2020. Applications of genetic programming to finance and economics: past, present, future. Genetic Programming and Evolvable Machines. 21, 33–53.

Brookhouse, J., Otero, F.E., Kampouridis, M., 2014. Working with opencl to speed up a genetic programming financial forecasting algorithm: initial results, in: Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 1117–1124.

Cavalcante, R.C., Brasileiro, R.C., Souza, V.L., Nobrega, J.P., Oliveira, A.L., 2016. Computational intelligence and financial markets: A survey and future directions. Expert Systems with Applications 55, 194–211.

Chen, T.l., Chen, F.y., 2016. An intelligent pattern recognition model for supporting investment decisions in stock market. Information Sciences 346, 261–274.

Chung, F.L., Fu, T.C., Luk, R., Ng, V., 2001. Flexible time series pattern matching based on perceptually important points. International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data , 1–7.

Frank, E., Hall, M.A., Witten, I.H., 2016. The WEKA workbench: online appendix for Data mining: practical machine learning tools and techniques. Morgan Kaufmann, Fourth edition.

Glattfelder, J., Dupuis, A., Olsen, R., 2011. Patterns in high-frequency FX data: discovery of 12 empirical scaling laws. Quantitative Finance 11, 599–614.

Guillaume, D.M., Dacorogna, M.M., Davé, R.R., Müller, U.A., Olsen, R.B., Pictet, O.V., 1997. From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets. Finance and stochastics 1, 95–129.

Kampouridis, M., Otero, F.E., 2017. Evolving trading strategies using directional changes. Expert Systems with Applications 73, 145–160.

Lin, T., Guo, T., Aberer, K., 2017. Hybrid neural networks for learning the trend in time series, in: Proceedings of the twenty-sixth international joint conference on artificial intelligence, pp. 2273–2279.

López-Ibánez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M., 2011. The irace package, iterated race for automatic algorithm configuration. Technical Report. Citeseer.

Özorhan, M.O., Toroslu, İ.H., Şehitoğlu, O.T., 2018. Short-term trend prediction in financial time series data. Knowledge and Information Systems , 1–33.

Poli, R., Langdon, W.B., McPhee, N.F., Koza, J.R., 2008. A field guide to genetic programming. Lulu. com.

Samanta, S., Pratama, M., Sundaram, S., Srikanth, N., . A dual network solution (dns) for lag-free time series forecasting .

Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K., 2013. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 847–855.

Tsang, E., 2010. Directional changes, definitions. Working Paper WP050-10 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex Revised 1, Tech. Rep. .

Tsang, E.P., Tao, R., Serguieva, A., Ma, S., 2017. Profiling high-frequency equity price movements in directional changes. Quantitative finance 17, 217–225.

Wan, Y., Gong, X., Si, Y.W., 2016. Effect of segmentation on financial time series pattern matching. Applied Soft Computing 38, 346–359.

Yin, J., Si, Y.W., Gong, Z., 2011. Financial time series segmentation based on turning points, in: Proceedings 2011 International Conference on System Science and Engineering, IEEE. pp. 394–399.