Artificial Intelligence Review (2026) 59:2
https://doi.org/10.1007/s10462-025-11419-z

A genetic algorithm for the optimization of multi-threshold trading strategies in the directional changes paradigm

Ozgur Salman[1] · Themistoklis Melissourgos[1] · Michael Kampouridis[1]

Received: 12 November 2024 / Accepted: 30 September 2025

Abstract
This paper proposes a novel genetic algorithm to optimize recommendations from multiple trading strategies derived from the Directional Changes (DC) paradigm. DC is an event-based approach that differs from the traditional physical time data, which employs fixed time intervals and uses a physical time scale. The DC method records price movements when specific events occur instead of using fixed intervals. The determination of these events relies on a threshold, which captures significant changes in price of a given asset. This work employs eight trading strategies that are developed based on directional changes. These strategies were profiled using varying values of thresholds to provide a comprehensive analysis of their effectiveness. In order to optimize and prioritize the conflicting recommendations given by the different trading strategies under different DC thresholds, we are proposing a novel genetic algorithm (GA). To analyze the GA's trading performance, we utilize 200 stocks listed on the New York Stock Exchange. Our findings show that it can generate highly profitable trading strategies at very low risk levels. The GA is also able to statistically and significantly outperform other DC-based trading strategies, as well as 8 financial trading strategies that are based on technical indicators such as aroon, exponential moving average, and relative strength index, and also buy-and-hold. The proposed GA is also able to outperform the trading performance of 7 market indices, such as the Dow Jones Industrial Average, and the Standard & Poors (S&P) 500.

Keywords Directional changes · Genetic algorithm · Trading strategies · Stock forecasting

✉ Themistoklis Melissourgos
themistoklis.melissourgos@essex.ac.uk

✉ Michael Kampouridis
mkampo@essex.ac.uk

[1] Centre for Computational Finance and Economic Agents, University of Essex, Wivenhoe, Colchester, Essex CO4 3SQ, UK

Published online: 11 November 2025

1 Introduction

Financial forecasting has made significant strides in stock investments in recent decades, particularly regarding return and risk. Modern portfolio theory, as introduced by Markowitz's seminal work (Markowitz 1952), sparked research in creating profitable portfolios for investors while also managing risk. Henceforward, forecasting of stock returns for traders has heavily evolved around two major techniques, namely, *Fundamental Analysis* (FA), and *Technical Analysis* (TA). Among the two, companies' financial statements serve as tools in the decision-making process for FA. In the fields of TA, historical prices and volume data are primary components. In this work, we focus on using TA, specifically from the perspective of novice traders who may lack extensive financial knowledge. This was motivated by the relative ease of use of the tools employed in the decision-making process.

The vast majority of research in the area of TA relies on predefined time intervals, such as daily, hourly, or weekly price data. However, this conventional time-based approach can lead to information loss, as it captures only specific snapshots of market activity while overlooking fluctuations occurring between those intervals. For instance, when relying on daily closing prices, only a single price per day is recorded, neglecting all intraday price movements that could provide valuable trading opportunities. This limitation affects both financial performance—where trading algorithms could otherwise capitalize on intermediate price shifts–and machine learning applications, where additional data points could enhance model training.

One possible solution is to incorporate higher-frequency data, such as hourly or minute-level prices, to reduce gaps between observations. However, this still imposes artificial constraints, as price changes occurring between these intervals remain unaccounted for. A more comprehensive alternative is to use tick-by-tick data, which records every market transaction or price change in real time. While this approach offers the most granular insight into market dynamics, it comes with significant trade-offs, including data storage requirements, computational complexity, and acquisition costs, making it less practical for many applications.

An alternative approach to such fixed time interval sampling methods is intrinsic time data sampling, which involves the sampling of data based on the occurrence of significant events in the market. The underlying concept is to record noteworthy market events that represent substantial price movements which would typically go unnoticed by traditional physical time sampling methods. Various intrinsic time sampling techniques have been documented, including "important points" (Pratt 2001), "perceptually important points" (Chen and Chen 2016), "turning points" (Yin et al. 2011), "zigzag" (Özorhan et al. 2019), and more recently, *directional changes* (DC) (Glattfelder et al. 2011; Tsang et al. 2017; Rostamian and O'Hara 2022; Li et al. 2022).

Despite being conceived very recently, the DC paradigm has produced innovative ideas, notably including scaling discoveries and indicators that are uniquely discovered through DC. In order to capitalize on those advancements we use DC in this work. The DC paradigm encompasses a sampling methodology that captures discrete instances of historical data solely when a price alteration surpasses a pre-determined positive threshold value. The latter is a critical parameter expressed as a percentage, it is denoted by θ, and is set by the trader based on their own perception of what constitutes a significant price change. Such changes can manifest as either an increase or decrease in value.

This work presents a novel DC framework, which applies eight distinct DC-based trading strategies across ten different DC thresholds. Each threshold generates a unique even-based series, and information from each series and strategy is combined to form a more informed trading approach. To achieve this aggregation, we employ a genetic algorithm (GA), which is a bio-inspired algorithm mimicking an evolutionary process, to optimize the parameters across the multi-strategy, multi-threshold recommendations. We test the framework extensively on daily data from 200 stocks (datasets) listed on the New York Stock Exchange. This objectives of this work are: (i) to develop accessible, profitable DC-based strategies that require minimal financial knowledge, and (ii) to improve trading performance by applying GA optimization.

The remainder of this paper is structured as follows: In Sect. 2, we will provide a comprehensive background to DC, and in Sect. 3 we will present a detailed review of the relevant DC literature from an artificial intelligence and machine learning perspective. In Sect. 4, we will discuss the methodology employed in our experiment. In Sect. 5, we will detail the experimental setup. Section 6 will focus on the presentation and discussion of our results. Finally, in Sect. 7, we will present the conclusion of this paper.

2 Background information on directional changes

2.1 Definitions

Initially, we would like to highlight two crucial points that emerge after DC profiling the physical data: i) based on this threshold parameter, the entire historical data can be analyzed solely along two directions, namely, *uptrend (UT)* and *downtrend (DT)*; (ii) in these trends, we can only observe two events, namely, a *directional change (DC) event* and an *overshoot (OS) event*.

As pointed out in the seminal work (Tsang et al. 2017), in contrast to physical time, which samples data points at regular time intervals, the DC samples data points from their peak and trough. By employing a pre-determined *threshold* (percentage), it becomes possible to decompose the data using these distinct components. In a DT (resp. UT), a last low price (resp. high price) is continuously updated to the minimum (resp. maximum) of the two prices: the current price $p(t)$ and the last minimum (resp. last maximum). The last minimum and maximum in these trends are naturally called *extremum* and are denoted by p_{ext_ℓ} and p_{ext_h}, respectively. The confirmation of a DC event in DT (resp. UT) occurs when the absolute price change between $p(t)$ and the p_{ext_h} (resp. p_{ext_ℓ}), denoted by $\Delta p := |p(t) - p_{ext_h}|$ (resp. $|p(t) - p_{ext_\ell}|$), is at least as high as the given threshold. The region between two DC events defines an OS event, which usually is of non-zero length.

Figure 1 demonstrates an example of the formation of consecutive DC and OS events for $\theta = 6\%$. Each data point represented on the graph corresponds to a paired combination of time-step (t) and price (e.g., point $A = (t_A, p_A) = (0, 99.9)$). Suppose we have a financial product whose price starts at 99.9\$ at $t = 0$ and decreases to 98\$ at $t = 1$, then to 97\$ at $t = 2$, and finally, to 94\$ at $t = 3$. Since the price change is smaller than the pre-specified value of θ, we do not consider the time interval $0 - 3$ as a DC event. Although the price decrease continues, we only update p_{ext_ℓ} (i.e., at $t = 3$, the lowest price we experienced is 94\$). At $t = 4$, the price jumps to 98\$, but again, due to not seeing the significant price

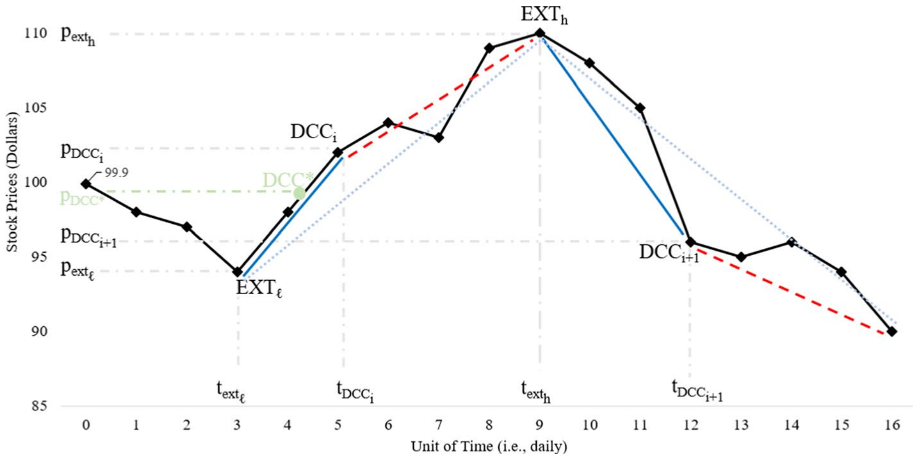

Fig. 1 Transformation of physical time data into the DC paradigm. The solid and dashed lines represent a set of events defined by a threshold $\theta = 6\%$, whereas the dotted lines correspond to events defined by a threshold $\theta = 17\%$. The solid and dotted lines represent the DC events, while the dashed lines indicate the OS events. For the threshold $\theta = 6\%$, there are two DC event confirmation points, at times 3 and 10. An uptrend takes place between the two extreme points, EXT_ℓ and EXT_h, which are confirmed retrospectively at their subsequent confirmation points, DCC_i and DCC_{i+1}

change that is defined by the θ, we still can not conclude a DC event. However, at $t = 5$, from p_{ext_ℓ} to our new price, Δp is at least as high as θ. In other words, within the interval from $t = 3$ to $t = 5$, a substantial price change of 6% is observed. Thus, we can conclude that an uptrend has occurred, and it is evident that the time duration of 3–5 qualifies as a DC event.

To detect the next DC event, this time we should observe a drop greater than the threshold's expected percentage. The event we are currently experiencing until this drop occurs is an OS event. Between $t = 6$ to $t = 7$, which is the first interval where we observe the price drop from $t = 5$ to $t = 9$, there is no DC event validation due to the drop being lower than θ (i.e., $|p(7) - p(6)| < \theta$). Meanwhile, p_{ext_h} keeps updating to the newest high. Therefore, when we reach $t = 9$, p_{ext_h} is at 110\$. From that point forward, we indeed observe a decrease at $t = 10$ and $t = 11$. However, these drops from the p_{ext_h} (110\$) are still not sufficient to conclude a DC event. At $t = 12$, we can observe that the required price change has occurred. Therefore, we can conclude that a DC event has taken place. Retrospectively, we also conclude that the OS in uptrend also occurred between $t = 5$ and $t = 9$. In this context, we would like to emphasize a point within the DC events profiled with the threshold $\theta = 17\%$ in Fig. 1 (indicated by dotted lines). While we expect that DC events are typically followed by OS events, it is essential to note that this pattern may not always hold true. DC events can occasionally be followed by another DC event in opposite trend due to data fluctuations.

Crucial to the definition of directional changes, are the notions of the extremum points (see EXT_ℓ, EXT_h in Fig. 1), and the *directional change confirmation point* DCC_i. As previously noted, an extremum point refers to the lowest price (resp. high price) in a DT (resp. UT). This point is continuously updated to reflect the minimum (resp. maximum) value between two prices: the current price and the last recorded minimum (resp. last maximum).

A confirmation point is a specific point in time at which one confirms the occurrence of a DC event. The interpretation of these points will be useful in our strategies' description in Sect. 4. Another important observation is that Δp can potentially be bigger than the minimum price change (determined by θ) required to identify it as a DC event. To account for this, the concept of a *theoretical confirmation point*, $DCC*$, is introduced. The theoretical confirmation point represents the hypothetical minimum or maximum price level required to confirm a directional change event, either a UT or a DT. It is important to note that the theoretical confirmation point may not actually exist or be encountered in the real market under most circumstances. Instead, it serves as a theoretical reference point used for analysis. This can be seen in Fig. 1, where a price change of 5.64$ from 94$ to 99.64$, which is exactly 6% more of the price at EXT_ℓ (recall that $\theta = 6\%$ in our example) between points EXT_ℓ and $DCC*$ is sufficient to confirm a DC event. The notation P_{DCC*} signifies the theoretical price that would be enough to conclude a DC event. Let us finally note that, as previously emphasized, DC paradigm encapsulates the entire given data through trends, namely, UT and DT. As an example from Fig. 1, the boundaries between EXT_ℓ and EXT_h represent UT, and from EXT_h to the upcoming EXT_ℓ will be DT. Algorithm 1 presents the pseudocode for generating DC events, which first appeared (using different notation) in Aloud et al. (2012b).

Require: Initialise variables (event is Downtrend event, $p_{ext_\ell} = p_{ext_h} = p(t)$, $t_{DC_{duration}}$ = physical time spent in a given DC event, t_{DCC} = specific DC confirmation time point in any trend

if Trend is Downtrend **then**
    if $p(t) \geq p_{ext_\ell} \cdot (1+\theta)$ **then**
        event $\leftarrow$ Uptrend
        $p_{ext_h} \leftarrow p(t)$ ▷ Price at the DC confirmation point for an Uptrend
        $t_{DCC} \leftarrow t$ ▷ End time for a Downtrend
        $t_{ext_\ell} \leftarrow t - t_{DC_{duration}}$ ▷ Start time for a Uptrend Overshoot Event
    else
        if $p_{ext_\ell} < p(t)$ **then**
            $p_{ext_\ell} \leftarrow p(t)$ ▷ Price at start of a possible Uptrend
else
    if $p(t) \leq p_{ext_h} \cdot (1-\theta)$ **then**
        event $\leftarrow$ Downtrend
        $p_{ext_\ell} \leftarrow p(t)$ ▷ Price at the DC confirmation point for a Downtrend
        $t_{DCC} \leftarrow t$ ▷ End time for a Uptrend
        $t_{ext_h} \leftarrow t - t_{DC_{duration}}$ ▷ Start time for a Downtrend Overshoot Event
    else
        if $p_{ext_h} > p(t)$ **then**
            $p_{ext_h} \leftarrow p(t)$ ▷ Price at start of a possible Downtrend

Algorithm 1 Pseudocode for generating DC events given threshold (θ)

Lastly, in this section, it is worth mentioning what we perceive as intersections between DC and TA. TA evaluates financial product movements through price trend analysis and graphical representation. In this work, we also rely on this point. TA indicators development follows predefined rules, yet interpretation varies among traders. Hence, it's important to bear in mind that indicators should not be regarded as inflexible, steadfast trade rules.

Instead, they possess a certain level of adaptability and are subject to the trader's discretion, affording a freedom and flexibility in their utilization. Thus, in this paper, our goal is to develop TA-like, DC-based trading strategies resembling widely used TA strategies.

3 Literature review

Technical analysis is widely utilized in financial markets to identify trading opportunities based on price movements and historical patterns. Over the years, technical indicators have been integrated into machine learning models to enhance predictive accuracy in algorithmic trading. Since the 1980 s, artificial neural networks have been applied to financial forecasting, leading to the development of more sophisticated AI-driven trading systems.

Several studies have investigated the combination of technical indicators with different predictive models. For example, Mostafa (2010) applied linear models alongside technical analysis indicators, while Nelson et al. (2017) employed long short-term memory (LSTM) networks to predict stock price trends. In Kamara et al. (2022), the authors introduced a hybrid deep learning framework incorporating technical analysis for financial forecasting, using two stocks as case studies. Additionally, Ghasemzadeh et al. (2024) emphasized the value of meta-synthesis techniques in financial systems, demonstrating how integrating traditional financial metrics with AI-based methods can improve decision-making. Similarly, Sharma and Verma (2024) explored the impact of technical indicators on deep learning models for option price prediction, reporting enhanced accuracy.

The use of evolutionary algorithms and particularly genetic algorithms also dates back to several decades ago (Brabazon et al. 2020). Recent works include Macedo et al. (2020), which demonstrated that genetic algorithms (GA) can optimize technical trading strategies by identifying market inefficiencies and improving profitability. In addition, Ito et al. (2020) used an evolutionary model that aggregates traders to predict stock returns using interpretable alpha factors, addressing market uncertainty and model adaptability. Further advancements include de Almeida and Neves (2022), which introduced self-adaptive evolutionary algorithms for stock market prediction and portfolio management, achieving higher Sharpe ratios and lower risk. In a related study, Long et al. (2023) combined genetic programming with directional changes and technical indicators within a multi-objective optimization framework, successfully balancing return and risk in trading strategies.

While all of the above works focus on physical time to summarize data, an alternative is summarizing data based on events, i.e. on intrisic time. As previously mentioned, directional changes is such a technique. Its origins can be traced back to Guillaume et al. (1997), which aimed at analyzing trend behavior. Since then there has been many studies that have been using directional changes. Early works focused on studying market dynamics under DC, e.g. Glattfelder et al. (2011) discovered 12 scaling laws that could hold for 12 foreign exchange currrency pairs. Other works also looked at creating DC-based indicators, e.g. Tsang et al. (2017); Tsang and Chen (2018) was one of the first to propose novel indicators in this domain. Later on, Tao (2018) build on the previous two works, by providing a new set of DC-based indicators. More recently, artificial intelligence and machine learning algorithms have been applied to DC to assist with the decision-making process during trading.

Early works on artificial intelligence have used agent-based modeling to simulate financial markets under the DC paradigm (Aloud et al. 2012a; Aloud 2016). An agent-based approach was also used by Bakhach et al. (2016) and Bakhach et al. (2018), the latter using the well-known C4.5 algorithm to predict whether future trades can be profitable.

The first work that used a machine learning algorithm in the DC framework was Gypteau et al. (2015), which applied genetic programming to combine trading actions under a single tree. Recently, DC-based trading strategies were proposed alongside reinforcement learning (Rayment and Kampouridis 2023, 2024). Meanwhile, the incorporation of DC features, such as trend reversals, into classification tasks has also led to the development of works utilizing Forex data (Adegboye et al. 2021; Adegboye and Kampouridis 2021; Adegboye et al. 2022; Rayment et al. 2023). Lastly, there has been works that have combined technical analysis and DC indicators, which has often resulted in improved trading performance when compared to only technical analysis or only directional changes (Long et al. 2022, 2023; Long and Kampouridis 2024). Such works have included the use of NSGA-II (Deb et al. 2002), a well-known multi-objective optimization algorithm that is based on genetic algorithms.

In simple terms, a *trading strategy* is a plan designed to facilitate the buying, selling, or holding of assets such as stocks, bonds, commodities, or intellectual property, with the ultimate objective of generating profit. The literature reveals that strategies based on DC, which can be easily implemented by traders with limited financial analysis expertise, are relatively scarce. Among the researches built on simple trading strategies, the first by Salman et al. (2022) implemented seven strategies derived from scaling discoveries and indicators from DC. Each strategy generated a distinct set of recommendations—Buy, Sell, or Hold—and the information from these individual recommendations was aggregated to produce a more informed overall recommendation by the use of a genetic algorithm. Subsequent work by Salman et al. (2023) focused on obtaining more comprehensive insights by introducing multiple distinct thresholds (θs) and again optimize them by a GA.

However, in the first work, the aggregated recommendation from the seven strategies was conducted using only threshold ($\theta = 2.5\%$), thereby constraining each trading strategy to the information provided by that specific DC threshold. This presents a significant challenge, as it is difficult to determine which DC threshold yields the most informative summary. In the second study, three strategies were aggregated using multiple thresholds, meaning that the overall decision was based solely on the insights from different thresholds. This, in turn, limited the range of information that could be provided by the different strategies.

In our current work, to overcome the aforementioned limitations, we will use different DC-based strategies, as well as multiple thresholds. As mentioned earlier, we will use a genetic algorithm to assist with the decision-making behind the potentially conflicting recommendations of the different trading strategies. The next section discusses in detail the methodological steps in our work.

4 Methodology

In this work, we propose a new model for optimizing trading strategies within the DC paradigm. Specifically, we construct a more fine-grained optimization via a genetic algorithm (GA), which now employs chromosomes that encapsulate *sub-strategies*, each with its own threshold. We call the current model *Multi-Strategy/Threshold-Genetic-Algorithm-Model*

(MSTGAM). This is going a step further than the previous works; unlike the model proposed in Salman et al. (2022) whose optimization focused only on a single threshold, or that proposed in Salman et al. (2023) where the optimization was applied only on thresholds, our current model aims to explore the GA optimization over sub-strategies.

A sub-strategy combines a trading strategy with a DC threshold θ, and we generate a set of sub-strategies by pairing each of our trading strategies (St1,..., St8) with corresponding thresholds ($\theta_1, ..., \theta_{10}$).[1] Each sub-strategy is represented as a gene in a chromosome and assigned a weight. Regardless of their weights, the genes recommend Buy, Sell, or Hold. The chromosome's decision-making process aggregates these weighted recommendations, with the total weight summing to 1. To resolve conflicting recommendations, we use a genetic algorithm (GA) to optimize the weights assigned to each sub-strategy.

The remainder of this section is structured as follows: In Sect. 4.1, we present the individual trading strategies. In Sect. 4.2 we discuss the thresholds selection. Finally, in Sect. 4.3 we demonstrate the GA methodology and how it is used to optimize the aforementioned trading strategies and thresholds.

4.1 Trading strategies

This section will explore the rationale and objectives of each trading strategy, summarized in Table 1. We present eight strategies, some based on indicators and others on scaling discoveries, all with the shared goal of applying TA-inspired methods within the DC framework. These strategies fall into two categories: adaptations of pre-existing strategies from the literature, modified for their initial application, or entirely novel strategies. We will structure the strategies by their distinct characteristics. In Sect. 4.1.1, we examine strategies

[1] There are 60 combinations of St1,.., St6 with $\theta_1, ..., \theta_{10}$, and 10 combinations of St7 and St8 with $\theta_1, ..., \theta_5$

Table 1 Execution signals as buy and sell

Strategy	Buy action	Sell action
St1	In DT, once the price change from p_{ext_h} reaches two θ	Same signal in the UT
St2	In DT, once the duration of its OS event reaches double the duration of DC event	Same signal in the UT
St3	In DT, once we see the $\mid OSV_{CUR} \mid$ is equal or greater than the $\mid OSVbest \mid$	Same signal in the UT
St4	In DT, once we see the $\mid TMV_{CUR} \mid$ is equal or greater than the $\mid TMVbest \mid$	Same signal in the UT
St5	In DT, once the duration of OS event over the DC event is equal or greater than *RD*	Same signal in UT
St6	In DT, once the randomly generated *p* is equal or greater than the *RN*	P_{DCC} in upcoming trend
St7	3rd consecutive OS in UT	P_{DCC} in DT
St8	3rd consecutive OS in DT	P_{DCC} in UT

1 and 2, grounded in scaling discoveries, which describe proportional relationships between physical quantities over significant intervals. In DC, these relationships connect price movements, duration, and frequency. Section 4.1.2 covers strategies 3 to 8, which use indicators, and concludes with Table 1, offering a concise summary of their execution mechanisms.

4.1.1 Strategies based on scaling discoveries

Scaling discoveries, in essence, explain the inherent connection between two physical quantities that exhibit proportional changes across a substantial range. In the context of DC, these associations primarily aim to establish mathematical links encompassing price movements, duration, and frequency. Among the 12 scaling discoveries found through DC (Glattfelder et al. 2011), two are highly important at connecting the DC and OS events by their average duration, and the price changes in each event.

The *first scaling law*, identified by Glattfelder et al. (2011), describes a recurring pattern where a directional change (DC) defined by a threshold (θ) is, on average, followed by an overshoot (OS) event with a price change approximately equal to that of the threshold θ. This relationship is captured in Eq. (1), where the symbol "$\approx$" denotes approximate equivalence.

$$\langle \Delta p_{DC} \rangle \approx \langle \Delta p_{OS} \rangle \approx \theta \tag{1}$$

Building on the scaling law, Strategy 1 (St1) involves purchasing a stock during a downtrend (DT) when a price change equal to or greater than twice the threshold θ is observed from its extremum point (i.e., p_{EXT_h}). It is crucial to note that if a price change of $2 \cdot \theta$ occurs at the confirmation point (DCC) from p_{EXT_h} (or p_{EXT_l} in the case of a sell), the buy (or sell) order is executed at the DCC. The same process is applied to initiate a sell order during an uptrend (UT). The logic behind this strategy is to capture the trend once the price change, as dictated by the scaling law, has occurred and to then wait for the opposite trend (i.e., UT) to realize a profit. Algorithm 2 provides an overview of how the trading strategy is structured.

```
if DC is in DT then
    if there is no open position and price change reaches 2·θ from p_EXT_h then
        buy one amount of share
    else
        Hold
else if DC is in UT then
    if there is an open position and price change reaches 2·θ from p_EXT_l then
        close the position by selling the share
    else
        Hold
```

Algorithm 2 Trading rule for St1

The *second scaling law* demonstrates a consistent pattern: on average, the duration of an OS event was approximately twice the duration of a DC event. Equation (2) highlights the scaling law, by aligning the notation of Glattfelder et al. (2011) let us denote by $\langle T_{OS} \rangle$ and $\langle T_{DC} \rangle$ the average time of an OS and DC event, respectively. Consequently, the previously mentioned scaling law can be expressed as follows:

$$\langle T_{OS} \rangle \approx 2 \cdot \langle T_{DC} \rangle, \tag{2}$$

Equation (2) underscores the scaling law, where the symbol "≈" denotes approximate equivalence again.

Strategy 2 (St2) follows this rationale: Upon observing a DC event, an execution signal is generated by checking the time duration of the DC and holding for double that time after the confirmation point *DCC*. Following this, a Buy order is executed if the market is in a downtrend (DT), or a Sell order is executed if the market is in an uptrend (UT) (If the position is already opened as a Buy). This strategy is designed to enable informed decision-making, based on the assumption that the scaling law holds for each distinct trend. Algorithm 3 provides an overview of the strategy's implementation.

```
if DC is in DT then
    if there is no open position and the time spent in OS is more than double the
time in its DC then
        buy one amount of share
    else
        Hold
else if DC is in UT then
    if there is an open position and the time spent in OS is more than double the
time in its DC then
        close the position by selling the share
    else
        Hold
```

Algorithm 3 Trading rule for St2

We believe that leveraging the above statistical properties provided by the scaling law can potentially yield profitable trading strategies, particularly due to their relative obscurity among traders. Therefore, the area of DC analysis presents a fertile ground for research, offering the potential for significant improvements in trading performance.

4.1.2 Indicator based strategies

This work introduces new DC-based indicators alongside existing ones for improved financial forecasting in traders' decisions. Note that here we only discuss the indicators used in the current work and in the most relevant recent work (Salman et al. 2022, 2023). For a more extensive exposition of indicators, we refer the reader to comprehensive sources such as Tao (2018).

Indicators

The indicators and their insights are as follows:

- Duration of DC events (T_{DC}): Total physical time spend in DC events.
- Duration of OS events (T_{OS}): Total physical time spend in OS events.
- Ratio of duration (*RD*): Total time spent in OS divided by total time spent in DC.

$$RD = \frac{T_{OS}}{T_{DC}} \tag{3}$$

- Number of DC events (N_{DC}): The total number of DC events throughout the investigated period.
- Number of Overshoot Events (N_{OS}): The total number of OS events in the profiled data.
- Ratio of number of events (*RN*):

$$RN = \frac{N_{OS}}{N_{DC}} \tag{4}$$

Notice, that $RN \in [0, 1)$, since in an extreme case it could be $N_{OS} = 0$, and in general, it also holds that $N_{DC} \geq N_{OS} + 1$, since there is at most one OS between two DCs.

- Theoretical Confirmation Point ($DCC*$): The earliest time after the extreme point (i.e., p_{ext_ℓ} or p_{ext_h} at which a price change equals θ in the direction opposite to the current trend. At the uptrend:

$$P_{DCC*} = p_{ext_\ell} \cdot (1 + \theta), \tag{5}$$

and at the downtrend:

$$P_{DCC*} = p_{ext_h} \cdot (1 - \theta). \tag{6}$$

- Overshoot Values at Current Points (OSV_{CUR}): The main goal of this indicator is to measure the magnitude of an OS event. It can be calculated as follows:

$$OSV_{CUR} = \frac{P_{CUR} - P_{DCC*}}{\theta \cdot P_{DCC*}}, \tag{7}$$

where P_{CUR} is the current price of the asset.

- Total Moves Value at Current Points (TMV_{CUR}): The main goal of this indicator is to measure total movement from the eyes of the leftmost extreme point. At uptrend,[2] it can be calculated as follows:

$$TMV_{CUR} = \frac{P_{CUR} - p_{ext_\ell}}{\theta \cdot p_{ext_\ell}}, \tag{8}$$

where P_{CUR} is the current price of the asset.***Strategies***
The following two strategies are based on the OSV_{CUR} (Eq. (7)) and TMV_{CUR} (Eq. (8)) indicators. The core idea behind their development is the dynamic utilization of the 'Best' values observed during the training phase, employing these values as execution triggers in the test set.

St3 focuses on the employment of the Overshoot Values at Current Points indicator. Within this strategy, we verify whether $|OSV_{CUR}| \geq |OSVbest_{DT}|$ in the test set. The

[2] Due to the fact that the usage of indicators is built upon absolute values, there is no difference whether we are in an uptrend or a downtrend. Fundamentally, the value of an indicator is the current price is determined by the extreme point price that forms the trend it is currently in.

way we determine our *OSVbest* which is used as threshold in their own way (a value that we decide upon for our trading mechanism) is as follows. Initially, we generate two distributions from the DC-profiled dataset as per Eq. (7): for every price in OS events in downtrends and uptrends. Therefore, if there is no OS events such that consecutive DC events occur, indicator values are not calculated for that part. These values are then divided into quartiles, each containing a median OSV_{CUR} value, resulting in four indicator values for both trends. Ultimately, the most favourable OSV_{CUR} values is identified through assessment, one for downtrend one for uptrend, denoted as *OSVbest*. This assessment conducted through testing these values by which of them generates the highest sharpe ratio in training set when we use the trading strategy that previously explained. Consequently, we identify two distinct *OSVbest* values: $OSVbest_{DT}$ for downtrend and $OSVbest_{UT}$ for uptrend.

In instances where this rule is satisfied, we examine the direction of the trend as a signal. If the trend direction is deemed as a downtrend (DT), we initiate a stock purchase and await to see the $|OSV_{CUR}| \geq |OSVbest_{UT}|$ in any upcoming uptrend (UT). In St3, our goal is to detect the trend reversal by observing when the indicator value reaches a certain magnitude. This approach allows us to capitalize on the uptrend shift by purchasing stocks at a lower price and selling them at a higher value. Algorithm 4 provides an outline of the process involved in constructing the trading strategy.

```
if DC is in DT then
    if there is no open position and |OSV_CUR| > |OSVbest_DT| [See Equation 7]
then
        buy one amount of share
    else
        Hold
else if DC is in UT then
    if there is an open position and |OSV_CUR| > |OSVbest_UT|  then
        close the position by selling the share
    else
        Hold
```

Algorithm 4 Trading rule for St3

St4 is founded upon the utilization of the Total Moves Value at Current Points indicator, as outlined by Eq. (8). In the formulation of this strategy, we once more adhere to the condition of verifying whether the magnitude of $|TMV_{CUR}|$ exceeds that of $|TMVbest_{DT}|$, akin to the approach in St3. The methodology for determining $|TMVbest|$ follows a similar process; however, the distinction lies in the calculation of the current value, which is based on Eq. (8). Again, we find two “Best” values, one for downtrend and one for uptrend, denoted by $TMVbest_{DT}$ and $TMVbest_{UT}$, respectively. In the final phase, the trend is assessed once again, and if it is recognized as a DT, a buy order for the stock is executed. We then await the UT, to execute a sell order when the condition is matched again. Similar to the previous strategy, our aim here is to anticipate an uptrend shift upon the indicator reaching a certain magnitude. The distinction lies in the measurement of the TMV_{CUR} indicator, which evaluates the trend from its initial starting point, offering a comprehensive view of the movement’s total trajectory. Algorithm 5 shows how trading strategy is integrated.

if DC is in DT **then**
  if there is no open position *and* $|TMV_{CUR}| > |TMVbest_{DT}|$ [See Equation 8] **then**
    buy one amount of share
  else
    Hold
else if DC is in UT **then**
  if there is an open position *and* $|TMV_{CUR}| > |TMVbest_{UT}|$ **then**
    close the position by selling the share
  else
    Hold

Algorithm 5 Trading rule for St4

The next two strategies are constructed based on the idea of establishing a relationship between OS and DC within the duration of their connection, as well as considering the overall relationship between the number of observed OS and DC events.

Strategy 5 (St5) is based on the ratio of the total time spent in OS events divided by the total time spent in DC events. We buy the stock in a downtrend whenever we observe that the time duration of OS divided to its DC event time duration is equal or greater than our predefined ratio value. The calculation of this fixed ratio is based on Eq. (3). For instance, if the duration of any given OS event to its DC duration exceeds the specified ratio *RD*, we execute a stock purchase if the current trend is DT. Similarly, when the current trend is UT, we wait for the same ratio value to be observed and then sell the stock. Algorithm 6 provides a summary of the process involved in building the trading strategy.

if DC is in DT **then**
  if there is no open position *and* ratio of time spent in OS to its DC $\geq RD$ [See Equation 3] **then**
    buy one amount of share
  else
    Hold
else if DC is in UT **then**
  if there is an open position *and* ratio of time spent in OS to its DC $\geq RD$ **then**
    close the position by selling the share
  else
    Hold

Algorithm 6 Trading rule for St5

Strategy 6 (St6) follows a similar process to that of St5. In this case, we establish our predefined ratio by dividing the total number of OS events by the total number of observed DC events, as indicated in Eq. (4). However, in this instance, the decision to buy stocks depends on a ratio that must consistently fall within the range of 0 to 1. The developed strategy operates based on probability, taking this ratio into account. If the randomly generated number is equal or greater than the predetermined ratio (*RN* as described in Eq. (4)) in a downtrend, a stock position is initiated only on *DCC* points of downtrends. To sell the stock, we await the next confirmation point during a uptrend. The underlying idea behind this strategy is based on the principle that sampling all the data using the DC paradigm gives us a general insight. By taking the number of OS events relative to DC events as a threshold, we aim to capture a quick uptrend in price during downtrends, in an aim not to see OS events when this ratio is met. Nevertheless, the degree of randomness in this strategy depends on the number of OS events observed to DC events observed, with the ultimate goal of capturing the bull market.

```
r is a random variable sampled from the uniform distribution in [0, 1]
if DC is in DT then
    if there is no open position and r ≥ RN [See Equation 4] at every DCC point
for St6 then
        buy one amount of share
    else
        Hold
else if DC is in UT then
    if there is an open position and subsequent DCC confirmed then
        close the position by selling the share
    else
        Hold
```

Algorithm 7 Trading rule for St6

The final two strategies aim to emulate the important notion of TA. To explain this conceptually, we first need to discuss two TA indicators. These two are *support*, and *resistance*. As prices decrease, they become more appealing to potential buyers who have been waiting. Eventually, the demand will reach a level that matches the available supply, causing prices to stabilize and stop declining. This is known as support (Lo et al. 2000). Resistance is the opposite of support. Prices rise when demand exceeds supply. As prices increase, a tipping point is reached where selling pressure outweighs buying interest. Building upon these two primary indicators, support and resistance, we have developed two strategies based on the sequential occurrence of OS events within the same trend. Similar to the triangle patterns from TA, the presence of three consecutive peaks signaling a reversal in price direction, since these patterns indicate a saturation in the current price. Therefore, in our strategy that is built upon the absence of OS events, we have also established it based on the absence of an equal number of OS events.

Strategy 7 (St7) is based on the following idea. In general, during a DT there is a DC interval followed by an OS interval, which in turn is followed by the next DC interval (where the latter signals a UT). Similarly for a UT. It is important to recall here that there is a case where there is no OS interval between the DC intervals, since the definition allows it. Consider now a sequence of UT-DT-UT-DT-UT. If in all of the DTs there is no OS and in each and all of the UTs there is an OS (i.e., three OS intervals), then St7 prescribes to buy the stock. Once the stock purchase occurs, we subsequently wait for a confirmation point in DT and then sell the stock. Algorithm 8 represents the functionality of the strategy.

if DC is in UT **then**
  if there is no open position *and* in the sequence of UT-DT-UT-DT-UT there is no OS in DTs *and* 3^{rd} consecutive OS in UT **then**
    buy one amount of share
  else
    Hold
else if DC is in DT **then**
  if there is an open position *and* the next DCC point is observed in DT **then**
    close the position by selling the share
  else
    Hold

Algorithm 8 Trading rule for St7

Strategy 8 (St8) is symmetric to St7, where instead of detecting three OS intervals in UT we detect them in DT. In particular, consider a sequence of DT-UT-DT-UT-DT. If in all of the UTs there is no OS and in each and all of the DTs there is an OS (i.e., three OS intervals), then we buy the stock. Once the stock purchase occurs, we subsequently wait for a confirmation point in UT and then sell the stock. Algorithm 9 prescribes the actions of the strategy.

if DC is in DT **then**
  if there is no open position *and* in the sequence of DT-UT-DT-UT-DT there is no OS in UTs *and* 3^{rd} consecutive OS in DT **then**
    buy one amount of share
  else
    Hold
else if DC is in UT **then**
  if there is an open position *and* when observe the next DCC point in UT **then**
    close the position by selling the stock
  else
    Hold

Algorithm 9 Trading rule for St8

In summary, these strategies were derived from a combination of scaling discoveries and indicators from DC. By resembling TA-like approaches in DC, they aimed to provide insights into potential outcomes in the financial markets for practitioners, and their results will be covered comprehensively in Sect. 6.

Trading Rules

There are several constraints and considerations to be aware of in the trading process, and these are as follows: (i) a new position (i.e., executing a buy, or sell on a stock) cannot be opened if a position is already open; therefore, a position must be closed[3] before a new one can be opened, (ii) *short selling*[4] is not permitted, meaning that all opening positions must involve taking a long position on a financial product, (iii) each trade is subject to a transaction cost of 0.25% applied to the price of the product at the time of execution.

[3] our objective was to treat each trade within a given stock as a single investment, considering the period from the initial purchase to the subsequent sale as a unified investment horizon.

[4] For a broader perspective, the concept discussed in Davies (2021) can be examined from the recent events that have garnered significant public attention.

4.2 Thresholds

It is evident that a specific value of θ determines a unique set of DC and OS events. For example, selecting smaller thresholds leads to more frequent events, providing the opportunity for timely actions. On the other hand, larger thresholds detect fewer events but allow for potential actions in response to more significant price changes.

Consequently, this study aims to capture the range of events by optimizing multiple thresholds. Thresholds selection process can be outlined as follows: Firstly, to ensure that the thresholds do not closely resemble each other, we divided the range from 0.05% to 2.75% into 10 equal intervals. These intervals are defined as follows: 0.05 for the first, 0.35 for the second, 0.65 for the third, continuing in this sequence up to 2.75 for the final interval (i.e., $2.75\% - 0.05\% = 2.70\%$, and $2.70\%/9 = 0.3\%$, with each interval incrementing by 0.30% and reaching up to 2.75%). Subsequently, we randomly selected the threshold values from 10 different normal distributions, each with a mean (μ) equal to the midpoint of one of these intervals and a standard deviation (σ) of 0.1. For example, the second threshold would be randomly selected from the distribution $\mathcal{N}(0.35, 0.1^2)$. By doing so, the resulting thresholds for turned out to be: $\theta_1 = 0.098\%, \theta_2 = 0.22\%, \theta_3 = 0.48\%, \theta_4 = 0.72\%, \theta_5 = 0.98\%, \theta_6 = 1.22\%, \theta_7 = 1.55\%, \theta_8 = 1.70\%, \theta_9 = 2\%,$ and $\theta_{10} = 2.55\%$.

4.3 Genetic algorithm optimization

4.3.1 Overview

Genetic algorithms are used in order to optimize complex objective functions. They do so by mimicking natural selection and genetics principles, and they are popular due to their time efficiency in generating high-quality solutions (Holland 1992).

As a short overview how the process is conducted for GA: (i) *Initial population*, start with a randomly generated potential solutions called chromosomes. Each chromosome contains genes representing different solution components. (ii) *Fitness evaluation*, Each chromosome's effectiveness (as a solution) is measured by a fitness function. In this work, we have chosen Sharpe Ratio, Eq. (9), as our fitness function; it balances profit and risk, and is a vital metric in financial markets. (iii) *Selection of chromosomes*, is conducted through methods that favors the fittest chromosomes among the current population. In our case, we used *tournament selection*, where k chromosomes.[5] are selected uniformly at random from the current population. Then, from the fittest of these k chromosomes, one with the highest fitness is picked to serve as a *parent chromosome*. (iv) *Operators*, among the possible two operators, we apply the operation of *crossover* with probability of p, where we have to pick a second parent by running another tournament selection of size k (with the first parent included in the pool of chromosomes), and create an offspring that will participate in the new population by replacing a strip of the first parent's genes with that of the second parent (see Sect. 4.3.3 for more details on the crossover process). With a probability of $1 - p$, we apply the operation of *mutation*, where this parent's genes undergo some random changes to create an individual that will participate in the new population. (v) *elitism*, where one of the fittest chromosomes—in our case, the chromosome with the highest Sharpe Ratio—pre-

[5] k is called *tournament size*.

served to the next population. In order to ensure the survival of a certain number of the fittest individuals from the current generation to the next. (vi) *Termination condition*, we repeat steps ii to v until a termination condition—in our case, certain number of generations—is met.

In this paper, we aim to determine whether overall performance can be improved by simultaneously applying multiple strategies and making various thresholds accessible to all strategies. Using GA, we optimized the recommendations generated by each strategy and the DC data profiles across different thresholds. The MSTGAM model enhanced strategy recommendations through a nature-based evolutionary algorithm, resulting in fitter chromosomes and, consequently, higher profits compared to individual strategies and benchmarks. In the following two subsections, we will discuss the process of constructing individuals by incorporating the strategies and thresholds discussed earlier. Additionally, we will explore the fundamental components and operations of the GA employed in our research.

4.3.2 Chromosome representation and action recommendations in GA

The MSTGAM algorithm optimizes the combination of multiple trading strategies, each operating under different DC thresholds, using the GA. Each strategy-threshold pair (sub-strategy) independently generates trading recommendations—Buy, Sell, or Hold– at any given time step. Since these sub-strategies may provide conflicting recommendations, MSTGAM employs a weighting mechanism to resolve these conflicts and determine the final action. At each time step, MSTGAM follows these steps:

1. Generate Individual Strategy Recommendations: Each sub-strategy (i.e., a combination of a trading strategy and a DC threshold) produces a recommendation.
2. Weight the recommendations: The GA assigns a weight to each sub-strategy, where the sum of all weights equals 1. These weights are optimised during training to maximise the Sharpe ratio.
3. Aggregate weighted recommendations: The total weight assigned to each action (Buy, Sell, or Hold) is calculated by summing the weights of the sub-strategies that recommend that action.
4. Execute the action with the highest weight: The action with the highest cumulative weight is executed in the market. To encourage trading activity, if at least two sub-strategies recommend an action other than Hold, MSTGAM ignores Hold recommendations and chooses between Buy or Sell based on their cumulative weights.

Below, we present the representation of the chromosomes in the GA population, followed by how MSTGAM combines signals from the various sub-strategies to arrive at a final trading decision.

Chromosome representation

In a GA, chromosomes are typically represented as a string of numbers, each in a particular range of values depending on the problem at hand; here, their domain is [0, 1]. Table 2 shows an example of a chromosome with only eight genes. Each cell in the string repre-

Table 2 An example of a chromosome representation with eight genes

0.045	0.001	0.450	0.102	0.130	0.050	0.015	0.207

sents a weight and corresponds to a variable to be optimized. In the GA, the population is initialized with random individuals, wherein each gene is assigned an initial value sampled uniformly at random from [0, 1].

Action recommendations

Building upon the generic GA representation, let us now discuss how the chromosome is utilized in our research. Firstly, a chromosome of MSTGAM consists of 70 distinct genes. Out of these 70 genes, 60 genes represent the 6 strategies, namely St1, St2, St3, St4, St5, and St6, each combined with 10 different thresholds $\theta_1, \ldots, \theta_{10}$ (see Sect. 4.2 for their values), and the remaining 10 genes correspond to St7, and St8, each combined with thresholds $\theta_1, \ldots, \theta_5$. The reason for using only 5 thresholds in the last two strategies is that, due to their specific construction, these strategies may have limited—or even, no—trade opportunities. A gene, i.e., a combination of trading strategy and threshold, is called a sub-strategy: St$i\theta j$ denotes trading strategy $i \in \{1, \ldots, 8\}$ under threshold $j \in \{1, \ldots, 10\}$. By focusing on a smaller value of thresholds, we can explore the unique characteristics and behaviors of these strategies within a more constrained parameter space.

Since a single trading strategy can be a part of many sub-strategies, it is possible for different trading strategies to provide conflicting recommendations of action. For example, St3θ2 might recommend Buy while St3θ5 might suggest Sell at the same point in time. To mitigate this issue, we introduce a weighting mechanism for them. At any given time in the price data of the training set, our GA will work to optimize 70 genes, which represent 70 distinct sub-strategies. For visualization purposes, in Table 3 we present a toy-example with a chromosome consisting of only 8 genes (sub-strategies).

From Table 3, at each point in time, sub-strategy will make a recommendation to MSTGAM on whether to buy, sell, or hold the current position. In the particular example, the individual recommended actions of the sub-strategies St1θ1, St2θ1, St4θ1, St5θ1, St6θ1, St7θ1 are to hold the stock at that given time, while the recommendation for St3θ1 is to buy, and for St8θ1 is to sell. In order to decide which action we take, we sum up the weights of the genes that recommend the same action, i.e., the sum of buying is 0.45; the sum of selling is 0.207; the sum of holding $0.045 + 0.001 + 0.102 + 0.130 + 0.050 + 0.015 = 0.343$. Then, the action that the entire chromosome will perform is the one that has the highest cumulative weight. In this example, buying the position has the highest weight sum with 0.45, therefore, at that specific time, the decision of the chromosome would be to buy the position. In general, the GA process optimizes the weights associated with individual sub-strategies to maximize the fitness function, which serves as a measure of the overall performance of the 70 recommendations.

However, in our experiments, the above approach resulted in a problematic situation appearing often: the large majority of the chromosome recommendations within most of the generations would be Hold. Therefore, to promote responsiveness, we implement a slight modification of that approach, which encourages a higher frequency of trades by artificially assigning a higher weight to Buy or Sell actions: if at any given time slot and chromosome, we observe more than two genes recommending anything other than Hold, we disregard the

Table 3 The chromosome representation includes 8 sub-strategies for θ1, their corresponding recommendations i.e. hold:0, buy:1, sell:2, and the hypothetical weights assigned to each recommendation

Sub-strategy	St1θ1	St2θ1	St3θ1	St4θ1	St5θ1	St6θ1	St7θ1	St8θ1
Action	0	0	1	0	0	0	0	2
Weight	0.045	0.001	0.450	0.102	0.130	0.050	0.015	0.207

Hold-genes, and decide the chromosome's recommendation according to the other genes' weights.

4.3.3 Operators, fitness function, and metrics

Here, we establish the operators and the fitness function employed within the GA framework. We employ a two-point crossover[6] operator with a probability p and a one-point uniform mutation operator with a probability of $1 - p$. Additionally, we incorporate *elitism*, which involves preserving the best chromosome from one generation to the next.

To evaluate the fitness of chromosomes, we utilize the Sharpe ratio (SR) as our fitness function. The SR accounts for risk-adjusted returns and is calculated using the following equation:

$$SR = \frac{R_p - R_f}{\sigma_p} \tag{9}$$

where R_p is the total rate of return calculated by summing the profits and losses for the entire duration of a given dataset., R_f is a risk-free asset, which is selected as 2.5% for a two-year dataset to preserve the resemblance of USA government bonds, and σ_p is the standard deviation of returns, i.e. the risk of the trading strategy.

Here, we would like to introduce the metrics that will be used in our performance analysis in the upcoming Sect. 6. The first one is Rate of Return (RoR) which is used to measure the profitability of an investment over a specific period. RoR is expressed as a percentage and the formula for calculating is as follows:

$$RoR = \frac{P_{t_{i+1}} - P_{t_i}}{P_{t_i}} \tag{10}$$

where, $P_{t_{i+1}}$, and P_{t_i}, represent the prices that we sell, and buy, respectively. For the risk metrics, we utilize two of them. The first one is Value at Risk (VaR), and its equation is as follows:

$$VaR_\alpha(P) = -F_P^{-1}(\alpha) \tag{11}$$

where $VaR_\alpha(P)$ represents the Value at Risk at a confidence level of α (i.e., 95% in our research) for an investment P. $-F_P^{-1}(\alpha)$ represents the inverse cumulative distribution function (quantile function) of the investment's return distribution evaluated at α. The negative sign is due to the fact that we are considering the lower tail of the distribution. The next metric is Standard Deviation and its calculation is as follows:

[6] In GA, the two-point crossover operation entails the selection of two random positions along two parental chromosomes. Subsequently, the genetic material located between these positions is exchanged, resulting in the creation of one offspring. In this work, we arbitrarily pick only one of the two offspring to participate in the next population.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{12}$$

where σ is standard deviation, N is the total number of trades, x_i is individual trade, and the μ is the average of the trades return.

Lastly, we also consider the Turnover Rate (ToR) and its calculation is as follows:

$$TurnoverRate = \frac{(Trades \times SharePrice)}{AveragePortfolioValue}. \tag{13}$$

Where *Trades* is the average number of trades, for a specific trading strategy, *SharePrice* is the average closing price of a stock over a given period, and *AveragePortfolioValue* is the average capital invested in the portfolio over a given trading period. As we are not performing portfolio optimization in this work, we are assuming an equal weight in the portfolio for each stock, and an initial capital of £100,000. The denominator (*AveragePortfolioValue*) represents the capital deployed over time, ensuring turnover is normalized for portfolio size. This formulation captures the frequency and magnitude of trading activity within each strategy, offering us insights into execution intensity.

5 Experimental setup

In this study, we analyze 200 publicly traded stocks listed on the New York Stock Exchange over the period from November 27, 2009, to November 27, 2019. The dataset consists of daily closing prices for each stock, obtained from Yahoo Finance using the “yfinance” Python module. The selection of these stocks was performed using the “random” module from the broader number of tickers. The reason for selecting 200 stocks is due to time efficiency considerations. Given the extensive number of tests required, a larger number stocks would be impractical, thus, 200 tickers were randomly chosen. The data set for each stock is divided into three parts: 56% for training, 24% for validation, and 20% for testing purposes. The validation set is utilized for parameter tuning of the GA, a topic that will be covered more comprehensively in the upcoming section. After tuning, the training and validation sets (comprising 80% of the total data) are combined to form a final training set, covering the first 8 years. In essence, we concatenate the validation set onto the training set to create the final training set, from which the results of the experiments in upcoming sections are derived. The selection of this specific period aims to exclude any potential distortions in the stock market data that could arise from the COVID-19 pandemic.

5.1 Parameter tuning

We performed a grid search to optimize the GA by fine-tuning the following parameters: population size, number of generations, and crossover probability p (mutation probability is equal to $1 - p$, hence to tuning was needed). To ensure reliable and robust results, we executed the GA 50 times for each combination of parameter values on 200 stocks. From each set of 50 runs, we retained the best chromosome (the one with the highest Sharpe Ratio). We applied the same procedure for selected parameter combinations and compared the results of their best chromosomes within the validation set. We also applied the Friedman non-parametric test to identify the best performing configurations. The optimized parameters resulting from the aforementioned procedure can be found in Table 4.

Consequently, we fix these parameters and run the GA for another 50 times on all of the 200 available stocks. Again, we pick one that performs the best, which is the final solution derived by our model's experiments (see Sect. 4.3 for a detailed exposition of the GA methodology).

5.2 Benchmarks

The aim of this study is to showcase that by utilizing a stochastic search technique, specifically, a GA, to optimize recommendations derived from multiple thresholds and strategies, we can improve trading performance beyond what has already been tested and documented in existing literature. Therefore, in Sect. 6, we wanted to examine the benchmark comparisons separately, dividing them into two categories: DC-based and Non-DC-based benchmarks.

5.2.1 DC-based benchmarks

Sub-strategies

We consider the strategies and the trade decisions they provide under different thresholds as individual strategies, as we explained as sub-strategies. Since one of the main goals of optimization is to derive these sub-strategies performance, we will use them as benchmarks in this chapter.

Eight Strategies Optimization on Particular Threshold

We have employed 8 strategies on a single threshold. The benchmark will be abbreviated by "MSGAM" *Multi-Strategy Genetic-Algorithm-Model*. To assess the consistency with our work in terms of used thresholds, we tested five thresholds, namely, 0.098%, 0.22%, 0.48%, 0.72%, 0.98%. To ensure the validity of the benchmark, we opted for the two best-performing results derived from the MSGAM model applied to two specific threshold ($\theta 1$, and $\theta 4$), and for the table spacing, they are abbreviated as $\mathrm{MS}_{\theta 1}$, $\mathrm{MS}_{\theta 4}$.

Different Thresholds Optimization on Individual Strategies

Here, we subjected each of our eight strategies to multiple threshold optimizations within their own contexts. we tested each 8 strategies with thresholds that were pointed out in Sect. 4.3.2, St1–St6 with 10 thresholds, St7 and St8 with 5 thresholds. The benchmark will be abbreviated to "MTGAM", *Multi-Threshold Genetic-Algorithm-Model*. To ensure the validity of the benchmark, we opted for the two best-performing results derived from the MTGAM model applied to two specific strategies, St7 and St8. In order to table spacing they are abbreviated by MT_7 and MT_8, respectively.

Executions on Confirmation Points

Table 4 Parameters

Population size	150
Number of generations	50
Tournament size	2
Crossover probability	0.95
Mutation probability	0.05

In this specific scenario, our trading approach involves executing trades immediately upon the confirmation of a directional change. Whenever we identify a trend as a downtrend, we initiate a buy at the confirmation point for the stock and then promptly sell it at the subsequent uptrend confirmation point. The benchmark will be abbreviated to “DCC”. The primary goal of this scenario is to assess trading profitability when focusing exclusively on DC events.

5.2.2 Non-DC Benchmarks

Technical analysis strategies

We use seven popular technical indicators. Based on these indicators, the parameter values for the employed strategies were set to values frequently observed in the field and the work by Achelis (2001). Their brief descriptions along with how these indicators are utilized in the execution processes of trading strategies as follows:

- **Average Directional Index (ADX)**: ADX quantifies price trend strength. Buy when ADX exceeds 25 upward trend. It is highly important to emphasize that the trends elucidated in the explanation of these indicators diverge from those discussed within the context of the DC paradigm. Sell when ADX surpasses 25 downward.
- **Aroon Indicator**: Identifies trends and their strength. Buy when Aroon Oscillator is positive (upward trend); sell when it’s negative (downward trend)
- **Commodity Channel Index (CCI)**: CCI identifies market trends. Buy signals occur when CCI is below -100, indicating oversold conditions, and sell signals when CCI is above 100, indicating overbought conditions.
- **Exponential Moving Average (EMA)**: Computes a 20-period EMA based on closing prices, emphasizing recent data with a designated alpha.[7] Buy when the closing price exceeds EMA (upward trend); sell when it falls below EMA (downward trend).
- **Moving Average Convergence Divergence (MACD)**: The MACD indicator is computed based on the 12-period and 26-period Exponential Moving Averages (EMAs) of closing prices. According to the MACD histogram, buy when below zero (potential upward trend), and sell when above zero (potential downward trend).
- **Relative Strength Index (RSI)**: The RSI is calculated over 14 periods, indicating overbought or oversold conditions. Buy signals are generated when RSI is below 30 (oversold), and sell signals when RSI is above 70 (overbought).
- **Williams %R (WilliamR)**: The WilliamR identifies overbought/oversold conditions. Buy signals occur at values below -80 (oversold), and sell signals at values above -20 (overbought).***Buy-and-Hold***

We also consider the BandH strategy as a benchmark, which involves purchasing and holding the product for a certain time without considering market fluctuations. In our model, the trader buys the product at the beginning of the test period and evaluates the performance monthly over the two-year period. Monthly returns are calculated after accounting for a transaction cost of 0.025%.

Market Indices

[7] Alpha represents a smoothing factor that determines how much weight is given ti the most recent data points.

To compare the performance and risk metrics of our model among the 200 stocks with the general movement of the stock market during our test period (November 27, 2017, to November 27, 2019), we used 7 market indices from the New York Stock Exchange. In our model, the trader buys the product at the beginning of the test period and evaluates the performance monthly over the two-year span. Monthly returns are calculated after accounting for a transaction cost of 0.025%. The indices are:

- Dow Jones Industrial Average (DJI): Represents 30 large, publicly-owned companies based in the USA.
- S&P 500 (GSPC): A market-cap-weighted index of the top 500 publicly traded U.S. companies.
- NYSE Composite Index (NYA): Encompasses all NYSE-listed common stocks.
- Russell 1000 Index (RUI): An index monitoring around 1,000 major U.S. equity market companies' performance
- Russell 2000 Index (RUT): A small-cap stock index covering the lowest 2,000 Russell 3000 Index stocks.
- Russell 3000 Index (RUA): An equity index representing the entire U.S. stock market, encompassing the top 3,000 U.S. companies.
- NYSE AMEX Composite Index (XAX): An index covering NYSE American-listed stocks, with a focus on smaller firms.

6 Results

This section of the paper showcases the outcomes of our experimental work. We start by presenting the DC-based and Non-DC benchmark strategies comparison to our MSTGAM results, followed by comparison of the market indices performances. As outlined in Sect. 5, we utilized daily closing prices as our data, covering a period of 10 years. Subsequently, the training phase encompassed the initial 8 years of data, and the results presented in this section are derived from the test set, covering the last 2 years of data.

We would like to remark that the goal of our work is twofold: (i) To create DC-based strategies that resemble technical analysis approaches, providing the reader with additional complementary power for making trade decisions, (ii) By optimizing these trading strategies with a combination of thresholds by the GA, improve the performance metrics over the single strategies as well as the financial benchmarks.

6.1 Summary statistics

Table 5 presents the average results of the Sharpe Ratio (SR), Rate of Return (RoR), Standard Deviation (STD), and Value at Risk (VaR), Number of Trades (Tra) for a set of 200 stocks across our MSTGAM (It will be abbreviated as "MST" for the table spacing) in comparison to DC related benchmarks. From the pool of 50 runs, the results presented in this section and subsequent section are based on a specific run. In this run, the chromosome with the highest Sharpe Ratio (SR) obtained during the training phase[8] is utilized in the test set. Evaluating the effectiveness of multiple runs is crucial; nevertheless, it is

[8] The performance results for the SR, RoR, and STD are from specific run (e.g., 10th run's chromosome).

Table 5 Average performance metrics based on 200 stocks in DC-based benchmarks, Sharpe Ratio (SR), Rate of Return (RoR), Standard Deviation (STD), Value at Risk (Var), Turnover Rate (ToR), and Number of Trades (Tra). Best value for each row is shown in bold

	MST	$MS_{\theta 1}$	$MS_{\theta 4}$	MT_7	MT_8	St1θ2	St2θ4	St3θ4	St4θ3	St5θ3	St6θ3	St7θ1	St8θ1	DCC
SR	**5.59**	1.07	1.71	3.12	2.37	−0.82	0.76	0.14	0.26	0.75	0.83	3.44	1.67	0.19
RoR	**0.22**	0.13	0.19	0.13	0.08	0.003	0.09	0.09	0.09	0.09	0.12	0.13	0.06	0.07
STD	0.04	0.09	0.1	0.03	0.03	0.06	0.09	0.1	0.1	0.08	0.07	0.03	**0.02**	0.05
VaR	0.05	0.12	0.12	0.03	**0.02**	0.09	0.13	0.1	0.1	0.12	0.1	0.02	0.02	0.06
ToR	**4.40**	0.70	0.57	0.58	0.49	1.99	0.70	0.37	0.50	0.74	1.25	0.57	0.42	2.52
Tra	70.19	10.27	8.61	8.61	7.02	27.48	10.17	5.42	7.24	10.66	18.2	8.36	6.07	35.9

equally vital to identify the optimal chromosome for practical application in real-world scenarios. This approach enables us to address realistic scenarios where traders would utilize the chromosome with the highest SR obtained during the training phase. For the MSGAM and MTGAM models, we utilized the two best-performing strategies for each model, determined by the highest Sharpe Ratio (SR). These strategies will be abbreviated as $\mathrm{MS}_{\theta i}$ and MT_i, respectively.

As explained in Sect. 5.2.1, we consider each different strategy and its resulting action set at each threshold as an individual strategy (sub-strategies). However, for the purpose of presenting the results within a limited space, we selected 8 of them for benchmark comparison. The selection of these 8 strategies was based on choosing the ones with the highest average SR performance.

Table 5 indicates that, on average, the MSTGAM strategy achieves the highest Sharpe Ratio (SR) with a value of 5.59, which is approximately 3.26 times higher than that of the MSGAM ($\mathrm{MS}_{\theta 4}$) model and 1.79 times higher than the MTGAM (MT_7) model. Compared to the DCC and individual strategies, St7θ1 ranks first. However, MSTGAM outperforms this strategy as well, with an SR approximately 1.65 times higher. As shown in the third column, MSTGAM delivers a rate of return (RoR) of 22%, adding an additional 3% over the next best-performing strategy, MSGAM. Additionally, risk metrics such as Value at Risk (VaR) and Standard Deviation (STD) were employed, alongside SR and RoR, to evaluate risk. From the last column, MSTGAM still maintains relatively moderate risk metrics, with an STD of 0.04 and a VaR of 0.09, despite executing a higher number of trades compared to the benchmarks. On the other hand, strategies St7θ1 and St8θ1, which are based on observing three consecutive OS events, displayed conservative VaR values at a 95% confidence level, driven by the lower number of trades they executed, both at 0.02. When considering the turnover rate (ToR), MST exhibits the highest turnover rate (4.40), suggesting that it is the most actively traded strategy. This aligns with its high number of trades (70.19), indicating frequent position adjustments. Despite the increased trading activity, MST's highest Sharpe Ratio (5.59) and Rate of Return (0.22), suggest that its frequent trades are effectively capturing profitable opportunities despite potential transaction costs.

In conclusion, the SR and RoR metrics demonstrate that (i) our approach is effective in generating profits, and (ii) the model's practical application in real-world trading can yield significant returns. However, the model shows some shortcomings in terms of risk management. Consequently, a key motivation for future research lies in enhancing the risk profile of these trading strategies.

Figure 2 presents a sample of the cumulative returns for MSTGAM and individual trading strategies across six stocks in the test set. Cumulative returns provide insight into strategy performance over time, enabling a clear comparison of long-term profitability and resilience across different market conditions. MSTGAM achieves the highest cumulative return at the end of the test period in four out of the six stocks presented. Notably, not all strategies exhibit the same trends as MSTGAM. For example, while MSTGAM shows an upward trend for the AAON stock, many other strategies either fail to exhibit a similar trend or do so on a much smaller scale. This highlights the differences in how each trading strategy adapts to varying market conditions. Additionally, in certain cases, some individual strategies surpass MSTGAM, suggesting that specific market conditions may favour alternative approaches. Nevertheless, the fact that MSTGAM generally achieves higher cumulative returns across

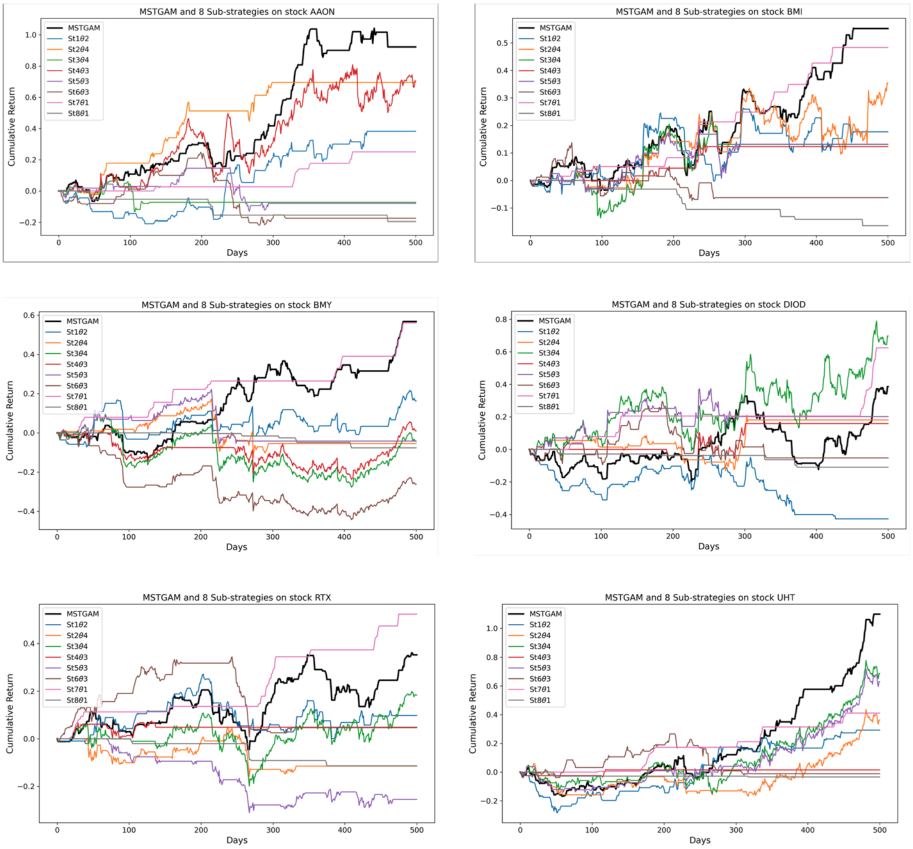

Fig. 2 Cumulative returns of MSTGAM and the 8 individual trading strategies

Table 6 Average performance metrics based on 200 stocks in Non-DC-based benchmarks, Sharpe Ratio (SR), Rate of Return (RoR), Standard Deviation (STD), Value at Risk (Var), Turnover Rate (ToR), and Number of Trades (Tra). Best value for each row is shown in bold

	MST	ADX	Ar	CCI	EMA	MACD	RSI	Wr	BandH
SR	**5.59**	−1.87	0.55	1.48	−2.64	−0.55	1.59	1.21	1.62
RoR	**0.22**	−0.03	0.07	0.09	0.01	−0.03	0.12	0.08	0.14
STD	0.04	0.1	0.07	0.07	0.06	0.07	0.1	0.07	0.1
VaR	0.05	0.12	0.14	0.16	0.05	0.15	0.16	0.16	0.13
ToR	4.40	6.26	1.24	0.89	2.31	1.31	0.42	0.86	1.63
Tra	70.19	5.59	17.68	12.78	31.97	17.8	6.15	12.31	24

the stocks in this figure,[9] along with its strong overall performance discussed in Table 5, indicates that it delivers a well-rounded, robust, and adaptable performance across different market conditions, reinforcing the effectiveness of its multi-threshold approach.

To assess MSTGAM's performance in comparison to mainstream Technical Analysis (TA) strategies, we compare it with several TA-based strategies in Table 6. A notable obser-

[9] A similar pattern is observed for other stocks as well.

vation is that MSTGAM achieves a Sharpe Ratio (SR) of 5.59, with the next highest SR being 1.62 from the Buy-and-Hold (BandH) strategy. Given that BandH involves holding a position on a monthly basis, MSTGAM's superior performance in terms of SR is particularly striking. Furthermore, MSTGAM delivers an additional return of over 8% in the RoR metric, which is another advantage. Another key point is that, while MSTGAM exhibits relatively higher SR and RoR compared to the TA-based strategies, its risk, as measured by various STD and VaR, remains below the benchmark values. When considering the turnover rate, ADX exhibits the highest value (6.26), despite having one of the lowest number of trades (5.59). This suggests that ADX executes fewer trades at a higher average price and/or operates with a lower average portfolio value, leading to a larger relative trading volume. However, ADX also reports negative Rate of Return (-0.03) and a negative Sharpe Ratio (-1.87), indicating that its high turnover does not translate into profitable performance. On the other hand, MST has a turnover rate of 4.40, which aligns with its high trade count (70.19). Unlike ADX, MST achieves the highest Sharpe Ratio (5.59) and Rate of Return (0.22), suggesting that its frequent trading is contributing positively to performance.

Figure 3 presents the box plot showing the distribution of values for the metrics across the 200 stocks. Focusing on MST's performance in the upper left section of the figure, the median value of MST is slightly above 5, with an average of 5.59, as detailed in Table 5. The low STD density indicates that the increase in the SR could be attributed to the risk-adjusted nature of the metric. Additionally, MST's median SR is significantly higher than that of other benchmarks. Conversely, in the RoR metric, MST is closely followed by the BandH strategy. However, in other box plots, the BandH strategy exhibits a lower median SR compared to other strategies, signifying lower risk-adjusted returns. This is due to its moderate RoR combined with higher volatility, as depicted in the STD plot. Analyzing the risk metrics in the lower box plots, MST's results are tightly clustered around a low median, which is distinct from other benchmarks. This suggests that MST effectively balances risk and return, demonstrated by its lower variance and competitive median values across these metrics.

To gain deeper insights into the results, we conducted the Friedman non-parametric statistical tests[10] under the assumption of the null hypothesis that all algorithms come from the same continuous distribution. In the Tables 7, 8, 9, 10, and 11, the second column displays the average rank of each algorithm (i.e., GA-optimized model, DC-based benchmark, or sub-strategies) while the third column presents the adjusted p-value obtained from the test comparing the average rank of each algorithm with that of the control algorithm (i.e., the algorithm with the highest rank). In adjusted p_{values}, we used the Post-hoc two-stage False Discovery Rate, abbreviated to *FDR* correction is employed to control the likelihood of making false discoveries (Type I errors) when conducting multiple pairwise comparisons.

Based on the observed results, it is evident that MSTGAM attains the highest rank and statistically outperforms all other algorithms at a significance level of $\alpha = 0.05$ in metrics SR and RoR. Tables 7 for SR and 8 for RoR show that MST ranks first in both metrics. Additionally, MST statistically outperforms every other benchmark at the significance level of $\alpha = 0.05$. In terms of SR, MT7 and MT8 follow in the ranks, while for RoR, the subsequent strategies are $\mathrm{MS}_{\theta 4}$ and BandH. Based on the results presented in Table 5, including the statistical results taht we found from these two table, we can conlude that MSTGAM

[10] These tests offer a robust and reliable means to assess differences among related groups. In this work, these groups are distributions from the related test metric results.

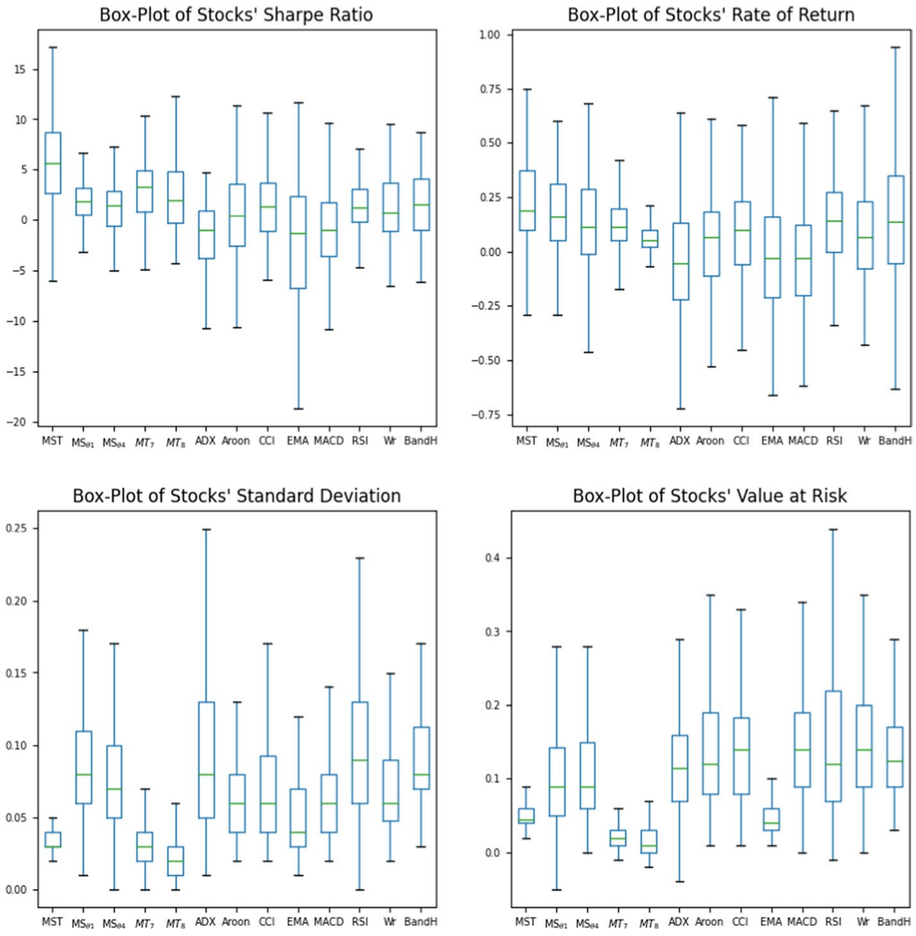

Fig. 3 Box-plots of MSTGAM (MST), preceding chapters-models, and non-DC related benchmarks results across 200 stocks on Sharpe Ratio, Rate of Return, Standard Deviation, and Value at Risk

outperforms both the DC-based and Non-DC based strategies. Importantly, MSTGAM also leads in performance metrics compared to strategies based on technical analysis.

In the risk metrics, STD and VaR, as shown in Tables 9 for STD and 10 for VaR, MSTGAM ranks third in STD and fourth in VaR. The top two positions in each metric are held by strategies St8 and St7, which are fed with multiple thresholds. Even though it doesn't rank first, it is noteworthy that MSTGAM closely follows the top strategies in the risk metrics, indicating its strong performance. In addition, in industry the focus is usually on aggregate metrics, such as Sharpe Ratio, rather than on individual metrics. Hence, it can be argued that significantly better performance in terms of SR compensates MSTGAM's Risk and VaR performance.

Table 12 presents the performance metrics for the proposed MSTGAM strategy (MST) against various market indices (DJI, GSPC, NYA, RUI, RUT, RUA, and XAX). The Sharpe Ratio (SR) is highest for MST (5.59), indicating strong risk-adjusted returns compared to the other indices. The Rate of Return (RoR) follows a similar pattern, with MST achieving

Table 7 The statistical test results for Sharpe Ratio were obtained using the non-parametric Friedman test, followed by the two-stage FDR correction to calculate adjusted p-values. Significant differences between the control algorithm (denoted with (c)) and the algorithms represented in a row at the α = 5% level are highlighted in boldface

Algorithm	Rank	Adjusted *p-value*
MST(c)	3.760	–
MT_7	**5.105**	8.843e−05
MT_8	**6.035**	2.210e−10
$\mathrm{MS}_{\theta 4}$	**6.450**	8.542e−14
RSI	**6.760**	1.236e−16
BandH	**6.830**	2.583e−17
CCI	**6.890**	7.284e−18
$\mathrm{MS}_{\theta 1}$	**7.140**	1.664e−20
Wr	**7.145**	1.523e−20
Ar	**7.450**	6.126e−24
EMA	**8.860**	1.400e−43
MACD	**8.995**	1.343e−45
ADX	**9.535**	1.537e−54

Table 8 The statistical test results for Rate of Return were obtained using the non-parametric Friedman test, followed by the two-stage FDR correction to calculate adjusted p-values. Significant differences between the control algorithm (denoted with (c)) and the algorithms represented in a row at the α = 5% level are highlighted in boldface

Algorithm	Rank	Adjusted *p-value*
MST(c)	4.81	–
$\mathrm{MS}_{\theta 4}$	**5.470**	2.734e−02
BandH	**6.115**	2.034e−04
RSI	**6.150**	1.516e−04
MT_7	**6.340**	2.120e−05
$\mathrm{MS}_{\theta 1}$	**6.460**	5.496e−06
CCI	**6.775**	8.568e−08
Wr	**7.185**	1.481e−10
Ar	**7.485**	7.315e−13
MT_8	**7.535**	3.153e−13
EMA	**8.570**	3.297e−23
MACD	**9.030**	1.046e−28
ADX	**9.075**	5.425e−29

the highest rate of return (0.22), while XAX is the only index with a negative rate of return (-0.001). In terms of risk, MST exhibits the lowest standard deviation (STD, 0.036), suggesting lower volatility relative to other indices. The Value at Risk (VaR) for MST (0.05) is also the lowest, implying lower downside risk. Regarding return distribution characteristics, MST has a slightly positive skewness (0.37), meaning returns have a slight rightward tilt, while other indices exhibit negative skewness, indicating a tendency toward larger losses. The kurtosis (1.42) for MST suggests a slightly fat-tailed return distribution, whereas other indices show lower kurtosis values, with XAX even exhibiting near-normal or platykurtic behaviour (-0.04). Overall, these statistics suggest that the MST algorithm delivers superior risk-adjusted returns with lower volatility and lower downside risk, distinguishing it from the benchmark indices.

Finally, we have examined the performance and risk metrics of the MSTGAM model for stocks, comparing their distribution to the performance of multiple strategies at a single threshold (MSGAM) and to the performance of models optimized using multiple thresholds (MTGAM). Specifically, we are utilizing the results from the best-averaged models, namely $\mathrm{MS}_{\theta 4}$, and MT_7 for comparison.

Table 9 The statistical test results for Standard Deviation were obtained using the non-parametric Friedman test, followed by the two-stage FDR correction to calculate adjusted p-values. Significant differences between the control algorithm (denoted with (c)) and the algorithms represented in a row at the $\alpha = 5\%$ level are highlighted in boldface

Algorithm	Rank	Adjusted *p-value*
MT_8(c)	1.940	–
MT_7	2.875	5.506e−02
MST	**2.920**	5.915e−06
EMA	**4.700**	7.851e−14
Ar	**6.730**	7.910e−52
MACD	**7.135**	3.657e−62
CCI	**7.660**	8.203e−77
Wr	**7.955**	1.599e−85
$MS_{\theta 1}$	**8.650**	1.679e−107
$MS_{\theta 4}$	**9.440**	9.549e−135
ADX	**9.455**	2.998e−135
RSI	**10.540**	1.782e−175
BandH	**10.940**	6.141e−191

Table 10 The statistical test results for Value at Risk were obtained using the non-parametric Friedman test, followed by the two-stage FDR correction to calculate adjusted p-values. Significant differences between the control algorithm (denoted with (c)) and the algorithms represented in a row at the $\alpha = 5\%$ level are highlighted in boldface

Algorithm	Rank	Adjusted p-value
MT_8(c)	2.150	–
MT_7	2.400	8.831e−02
EMA	**3.840**	3.632e−10
MST	**4.055**	5.698e−13
$MS_{\theta 4}$	**7.235**	2.010e−32
$MS_{\theta 1}$	**7.510**	3.218e−38
ADX	**8.200**	5.504e−53
RSI	**8.715**	2.932e−66
Ar	**8.900**	3.866e−70
BandH	**9.015**	4.745e−73
CCI	**9.320**	5.216e−82
MACD	**9.365**	8.081e−83
Wr	**9.510**	4.772e−87

Figure 4 presents the distributional characteristics of SR, RoR, STD, and VaR across different models. Regarding SR, MST demonstrates near symmetry (skewness: −0.039, kurtosis: 0.434), suggesting a balanced risk-adjusted return profile. MT7 is left-skewed (−0.459) with moderate kurtosis (1.878), implying frequent smaller gains but a risk of significant losses. MSθ4 exhibits slight right skew (0.179) and heavy tails (kurtosis: 4.527), indicating elevated risk. Across 200 stocks, MST tends to maintain a higher SR mean with a balanced risk-return trade-off. For RoR, MST exhibits mild right skewness (0.367) and moderate kurtosis (1.423), suggesting occasional larger gains. MT7 shows a more pronounced right skew (0.735) and heavier tails (kurtosis: 5.203), indicating higher gain potential with increased risk. Similarly, MSθ4 has a skewness of 0.768 and kurtosis of 2.040, reinforcing a right-skewed distribution. Overall, MST is centered around a higher RoR mean. For STD, all models—MST, MT7, and MSθ4—exhibit high skewness (2.787, 2.844, and 3.269, respectively) and extreme kurtosis (14.138, 12.080, and 15.110), indicating substantial right skewness, heavy tails, and a highly volatile risk profile. MST appears centered around a lower STD mean. In terms of VaR, MST has moderate right skewness (0.906) and light tails (kurtosis: 0.470), implying some risk of significant losses but less extreme outcomes. MT7

Table 11 The statistical test results for Turnover Rate were obtained using the non-parametric Friedman test, followed by the two-stage FDR correction to calculate adjusted p-values. Significant differences between the control algorithm (denoted with (c)) and the algorithms represented in a row at the $\alpha = 5\%$ level are highlighted in boldface

Algorithm	Rank	Adjusted *p-value*
ADX(c)	1.000	–
MST	**2.000**	3.48e−135
DCC	**3.000**	0.00e+00
EMA	**3.993**	0.00e+00
$\mathrm{St1}_{\theta 2}$	**4.993**	0.00e+00
BandH	**5.993**	0.00e+00
MACD	**6.980**	0.00e+00
$\mathrm{St6}_{\theta 3}$	**7.948**	0.00e+00
Ar	**8.300**	0.00e+00
CCI	**9.309**	0.00e+00
Wr	**10.180**	0.00e+00
$\mathrm{St5}_{\theta 3}$	**11.150**	0.00e+00
$\mathrm{St2}_{\theta 4}$	**12.051**	0.00e+00

Table 12 Performance and Risk metric comparison between MSTGAM (MST) and market indices (% for RoR). Best value highlighted by bold

	MST	DJI	GSPC	NYA	RUI	RUT	RUA	XAX
SR	**5.59**	4.15	4.20	1.96	4.08	1.39	3.63	−1.46
RoR	**0.22**	0.0076	0.0077	0.0039	0.0076	0.0038	0.0069	−0.001
STD	**0.036**	0.039	0.041	0.038	0.042	0.051	0.041	0.043
VaR	**0.05**	0.06	0.07	0.07	0.07	0.09	0.07	0.08
Skewness	**0.37**	−0.80	−0.95	−0.79	−0.92	−0.72	−0.89	−0.14
Kurtosis	**1.42**	0.45	0.94	0.73	0.96	0.61	0.94	−0.04

exhibits a pronounced right-skewed (2.484) and heavy-tailed (kurtosis: 8.951) distribution, signaling increased loss risk. MSθ4 is highly right-skewed (5.012) with extreme kurtosis (38.596), indicating a substantial risk of extreme losses. Overall, MST is associated with a lower mean VaR.

6.2 Discussion

One important consideration in our results is the relatively high Sharpe ratio observed in our proposed MSTGAM approach. While a Sharpe ratio above 5 may appear unusually high, it is noteworthy that this is observed only in MSTGAM, whereas the other algorithms and benchmarks in our study exhibit significantly lower values. This suggests that MSTGAM effectively optimizes the combination of trading strategies to enhance performance, rather than indicating that all tested approaches achieve such high returns relative to risk.

Additionally, we note that the transaction cost applied in our experiments is the same across all tested algorithms. While real-world trading environments may involve additional hidden costs, any underestimation in our model would apply equally to all methods and is therefore unlikely to be the sole factor contributing to MSTGAM's superior performance.

Furthermore, although our results are based on a separate test set, ensuring that the model is evaluated on unseen data, we acknowledge that market dynamics evolve over time. As such, performance may vary when applied to different time periods. It is worth noting that our study used data that spans from 2017 to 2019. While this period was chosen to provide

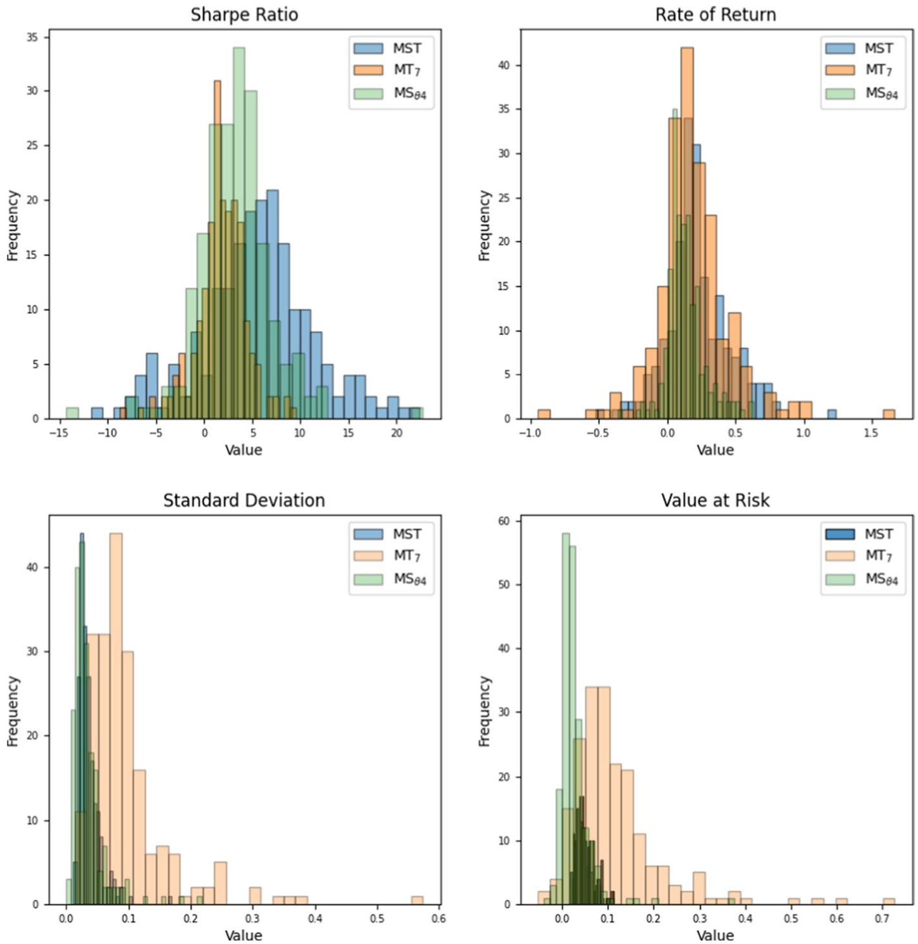

Fig. 4 Distribution of stocks performance and risk metrics, using MSTGAM (MST), MSGAM ($MS_{\theta 4}$), MTGAM (MT7)

a sufficiently long dataset for analysis, it avoids the extreme market conditions introduced by the COVID-19 pandemic. The volatility and structural shifts in financial markets during the pandemic were unprecedented and may not be representative of typical trading environments. While our findings remain valid for the examined period, future work could extend the analysis to include more recent data, particularly to assess the robustness of our proposed approach under extreme market shocks.

Furthermore, while MSTGAM incorporates risk awareness through the Sharpe ratio as its fitness function, it primarily focuses on return volatility rather than more sophisticated risk measures such as maximum drawdown or tail risk. It would thus be interesting in considering other types of fitness function. We discuss this in more detail in Sect. 7.

Finally, we recognize that results may differ when applied to other U.S. equities or foreign markets. However, our study does not claim universal applicability across all financial instruments. Instead, we report findings based on the selected datasets and time period

Table 13 Approximate computational times for models

Models	MSTGAM	MSGAM	MTGAM	Benchmarks
Model training	$\sim$ 60 mins	$\sim$ 30 mins	$\sim$ 30 mins	–
Estimation	$\sim$ 30 s	$\sim$ 30 s	$\sim$ 30 s	$\sim$ 20 s
Trading	$\sim$ 10 s	$\sim$ 10 s	$\sim$ 10 s	$\sim$ 5 s

examined in this paper. Given that our experiments include 200 different datasets, we believe the results demonstrate a degree of generalizability within this context. Nonetheless, further studies applying MSTGAM to different markets could provide additional insights into its adaptability.

6.3 Computational times

Table 13 shows the average computational times for MSTGAM, MSGAM, and MTGAM have been used in Sect. 6. As expected, due to the addition of more thresholds in the GA optimization phase, MSTGAM requires more computational time for the training phase compared to other benchmarks. Computations were conducted using the high-performance cluster at the University of Essex, which comprises a mix of Intel E5-2698, Intel Gold 5115, 6152, and 6238 L processors, each equipped with between 500GB and 6TB of RAM. It is important to acknowledge that the learning process on the training set typically occurs offline, rendering a duration of 55–60 min. Once training is successfully completed, its best chromosome is applied to the test set, requiring a mere 15 s for execution. Furthermore, it is important to note that parallelization techniques can be employed to reduce the computational time required for these algorithms (Brookhouse et al. 2014).

7 Conclusion

In conclusion, this paper introduces trading strategies that are created based on the DC paradigm. The experiments were conducted on the 200 stocks sourced from the NYSE. By incorporating GA optimization across various thresholds and strategies, we enhanced the performance of our strategy, namely MSTGAM, to a significant extent. In the comparative analysis, we examined two types of benchmarks: DC-based benchmarks and Non-DC based benchmarks (Technical Analysis strategies, BandH, and Market Indices). In the first scenario, MSTGAM demonstrated superior performance compared to its benchmarks, with the only exception being the risk metrics for MSGAM (derived from our previous research technique). Additionally, MSTGAM outperformed all selected Technical Analysis strategies, and BandH across all performance metrics. Finally, MSTGAM shows better performance in both performance and risk metrics compared to market indices based on the NYSE.

As explained earlier, one limitation of our study was the selection of the validation period, which spans from 2017 to 2019. Investigating the robustness of MSTGAM across varying market conditions is thus an important direction for future work.

In addition, future work could explore alternative fitness functions that integrate additional risk metrics or adaptive mechanisms to dynamically adjust trading strategies in response to changing market conditions. In particular, incorporating metrics such as maximum drawdown, conditional value at risk (CVaR), or downside deviation could provide a more comprehensive assessment of risk beyond return volatility. Additionally, adaptive mechanisms, such as reinforcement learning-based dynamic portfolio allocation, could enable the model to adjust its strategy in real time, based on evolving market conditions, enhancing both robustness and practical applicability.

Lastly, in future work we would like to explore research that focuses on identifying optimal thresholds. To achieve this, instead of relying on a range that adequately captures the daily changes in stock prices, we plan to conduct a distribution analysis using a larger number of thresholds. By doing so, we hope to gain insights into how we can improve our risk metrics and enhance their effectiveness.

Author contributions OS (PhD student) along with MK (main supervisory) conceived research project. OS conducted experiments and wrote first draft of the manuscript. MK and TM (second supervisor) reviewed and revised manuscript.

Funding This research received no external funding.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

Achelis SB (2001) Technical analysis from A to Z

Adegboye A, Kampouridis M (2021) Machine learning classification and regression models for predicting directional changes trend reversal in fx markets. Expert Syst Appl 173(114):645

Adegboye A, Kampouridis M, Otero F (2021) Improving trend reversal estimation in forex markets under a directional changes paradigm with classification algorithms. Int J Intell Syst 36(12):7609–7640

Adegboye A, Kampouridis M, Otero F (2022) Algorithmic trading with directional changes. Artificial Intelligence Review pp 1–26

Aloud ME (2016) Profitability of directional change based trading strategies: The case of saudi stock market. International Journal of Economics and Financial Issues 6(1)

Aloud M, Fasli M, Tsang E, et al (2012a) Modelling the high-frequency fx market: an agent-based approach. In: University of Essex, United Kingdom, Citeseer

Aloud M, Tsang E, Olsen R, et al (2012b) A directional-change event approach for studying financial time series. Economics 6(1):20120,036

Bakhach A, Chinthalapati VLR, Tsang EP et al (2018) Intelligent dynamic backlash agent: a trading strategy based on the directional change framework. Algorithms 11(11):171

Bakhach A, Tsang E, Ng WL, et al (2016) Backlash agent: A trading strategy based on directional change. In: 2016 ieee symposium series on computational intelligence (ssci), IEEE, pp 1–9

Brabazon A, Kampouridis M, O'Neill M (2020) Applications of genetic programming to finance and economics: past, present, future. Genet Program Evolvable Mach 21:33–53

Brookhouse J, Otero FE, 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
Chen TL, Chen FY (2016) An intelligent pattern recognition model for supporting investment decisions in stock market. Inf Sci 346:261–274
Davies R (2021) Gamestop: how reddit's amateurs tripped wall street's short-sellers. https://www.theguardian.com/business/2021/jan/28/gamestop-how-reddits-amateurs-tripped-wall-streets-short-sellers
de Almeida RL, Neves RF (2022) Stock market prediction and portfolio composition using a hybrid approach combined with self-adaptive evolutionary algorithm. Expert Syst Appl 204(117):478
Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: nsga-ii. IEEE Trans Evol Comput 6(2):182–197
Ghasemzadeh E, Keramati MA, Mehrinejad S et al (2024) Applying meta-synthesis techniques in identifying optimization components of fintech based on artificial intelligence indicators in the financial market. Int J Innov Manag Organ Behav 4(3):173–179
Glattfelder JB, Dupuis A, Olsen RB (2011) Patterns in high-frequency fx data: discovery of 12 empirical scaling laws. Quant Finance 11(4):599–614
Guillaume DM, Dacorogna MM, Davé RR et al (1997) From the bird's eye to the microscope: a survey of new stylized facts of the intra-daily foreign exchange markets. Finance Stochast 1:95–129
Gypteau J, Otero FE, Kampouridis M (2015) Generating directional change based trading strategies with genetic programming. In: Applications of Evolutionary Computation: 18th European Conference, EvoApplications 2015, Copenhagen, Denmark, April 8–10, 2015, Proceedings 18, Springer, pp 267–278
Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
Ito K, Minami K, Imajo K, et al (2020) Trader-company method: a metaheuristic for interpretable stock price prediction. arXiv preprint arXiv:2012.10215
Kamara AF, Chen E, Pan Z (2022) An ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices. Inf Sci 594:1–19
Li S, Tsang EP, O'Hara J (2022) Measuring relative volatility in high-frequency data under the directional change approach. Intell Syst Account Finance Manag 29(2):86–102
Lo AW, Mamaysky H, Wang J (2000) Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. J Financ 55(4):1705–1765
Long X, Kampouridis M (2024) α-dominance two-objective optimization genetic programming for algorithmic trading under a directional changes environment. In: IEEE Computational intelligence for financial engineering and risk (CIFER), IEEE
Long X, Kampouridis M, Kanellopoulos P (2022) Genetic programming for combining directional changes indicators in international stock markets. In: International conference on parallel problem solving from nature, Springer, pp 33–47
Long X, Kampouridis M, Kanellopoulos P (2023) Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators. In: 2023 IEEE Congress on evolutionary computation (CEC), IEEE, pp 1–8
Macedo LL, Godinho P, Alves MJ (2020) A comparative study of technical trading strategies using a genetic algorithm. Comput Econ 55(1):349–381
Markowitz H (1952) Portfolio selection*. J Financ 7(1):77–91
Mostafa MM (2010) Forecasting stock exchange movements using neural networks: empirical evidence from Kuwait. Expert Syst Appl 37(9):6302–6309
Nelson DM, Pereira AC, de Oliveira RA (2017) Stock market's price movement prediction with long short-term memory neural networks. In: 2017 International joint conference on neural networks (IJCNN), IEEE, pp 1419–1426
Özorhan MO, Toroslu İH, Şehitoğlu OT (2019) Short-term trend prediction in financial time series data. Knowl Inf Syst 61:397–429
Pratt KB (2001) Locating patterns in discrete time-series. Master's thesis, University of South Florida
Rayment G, Kampouridis M (2023) High frequency trading with deep reinforcement learning agents under a directional changes sampling framework. In: 2023 IEEE symposium series on computational intelligence (SSCI), IEEE, pp 387–394
Rayment G, Kampouridis M (2024) Enhancing high-frequency trading with deep reinforcement learning using advanced positional awareness under a directional changes paradigm. In: International conference on machine learning and applications
Rayment G, Kampouridis M, Adegboye A (2023) Predicting directional change reversal points with machine learning regression models. In: IEEE symposium on computational intelligence for financial engineering and risk (CIFEr)
Rostamian A, O'Hara JG (2022) Event prediction within directional change framework using a cnn-lstm model. Neural Comput Appl 34(20):17,193-17,205

Salman O, Kampouridis M, Jarchi D (2022) Trading strategies optimization by genetic algorithm under the directional changes paradigm. In: 2022 IEEE congress on evolutionary computation (CEC), IEEE, pp 1–8

Salman O, Melissourgos T, Kampouridis M (2023) Optimization of trading strategies using a genetic algorithm under the directional changes paradigm with multiple thresholds. In: 2022 IEEE congress on evolutionary computation (CEC)

Sharma A, Verma C (2024) Investigating the impact of technical indicators on option price prediction through deep learning models. 2024 IEEE international conference on electronics. IEEE, Computing and Communication Technologies (CONECCT), pp 1–5

Tao R (2018) Using directional change for information extraction in financial market data. PhD thesis, University of Essex

Tsang E, Chen J (2018) Regime change detection using directional change indicators in the foreign exchange market to chart brexit. IEEE Trans Emerg Top Comput Intell 2(3):185–193

Tsang EP, Tao R, Serguieva A et al (2017) Profiling high-frequency equity price movements in directional changes. Quant Finance 17(2):217–225

Yin J, Si YW, 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