Nosherwan KHAN, PhD (corresponding author)
nosherwan@cust.edu.pk
Capital University of Science and Technology, Islamabad, Pakistan

Arshad HASSAN, PhD
arshad@cust.edu.pk
Capital University of Science and Technology, Islamabad, Pakistan

Do Decimated Based Trading Model Outperform the Buy-and-Hold Investment Strategy

Abstract. *Trading Algorithms have significantly transformed the stock markets in recent years and practically replaced the technical analysts in internet trading to a considerable extent. However, due to the large computing requirements, these algorithms face issues in terms of speed and yield rates. To address these issues, this study introduces the algorithm, which uses decimated data to retain performance while reducing computational needs. This decimated method provides a viable solution to the computational issues encountered in online stock forecasting and devising a computerised trading strategy.*
The program follows stringent statistical criteria for selection of decimation rates, window size, and performance measures. Real-world stock market data from five well-known companies, notably Apple, Amazon, Alphabet, Meta, and Netflix, is used to assess the algorithm's efficacy in making investment decisions within a given time frame. The results indicate that the decimated algorithm has the potential to optimise stock returns in real time while demonstrating steady performance by using established concepts from information theory and making rational decisions about decimation rates. This research adds to the understanding of the dynamics of automated trading by shedding light on how to achieve optimal performance while minimising computational costs.

Keywords: *decimated data, online stock selection, trading algorithm, D-OLMAR algorithm.*

JEL Classification: E17, E47, F17.

1. Introduction

The debate over the efficiency of active versus passive investment management has a long history. Active investment management includes creating trading strategies based on specific trading rules built on historical patterns. Conversely, passive investment management involves Buy and Hold. With the growth of computerised trading and the use of algorithms for online trading, the field of portfolio management has experienced tremendous change. These algorithms have replaced traditional technical analysts and streamlined the stock trading process. However, they suffer from intrinsic computational efficiency issues, which can impair their speed and profitability.

DOI: 10.24818/18423264/58.1.24.15

B. Li and Hoi (2014) have summarised the creative strategies to enhance portfolio performance while minimising processing demands to address these issues. To achieve an ideal balance between performance and computational costs, the decimated algorithm uses the concept of decimated data and employs strong theoretical and statistical requirements, such as the decimation rates, window size, and selection of acceptable statistical criteria. Because of its potential to improve stock forecast and selection strategy, the decimated algorithm has attracted the attention of academics and practitioners. The program tries to maintain performance levels while lowering the computational strain involved in processing large amounts of stock market data by carefully selecting and decreasing the data set. This method offers a fresh viewpoint on investment optimisation by emphasising the necessity of data reduction in attaining efficient and effective decision-making processes.

The decimated algorithm is a potential method for effective stock selection and portfolio optimisation that makes use of data reduction techniques. This approach provides potential solutions to computational issues in stock selection and prediction by carefully employing decimated data. This article’s empirical evaluation and analysis contribute to the progress of investment decision strategies and provide vital insights into establishing efficient and effective decision-making processes in the setting of computerised trading.

2. Background of Algorithm Based Trading

In the financial markets, the application of machine learning algorithms in stock prediction, selection, and online trading has attracted a lot of interest of investment professionals. Researchers and practitioners identify several methods to improve the accuracy and efficiency of stock forecasting and stock selection. These algorithms are useful tools for financial experts who want to make informed stock market selections. Because of their ability to handle non-linear data, Support Vector Machines (SVMs) is frequently used in stock prediction.

SVMs are used to forecast stock market trends using numerous technical indicators as input characteristics (Gururaj et al., 2019; Kewat et al., 2017). These studies report promising prediction accuracy. Artificial Neural Networks (ANNs) are also widely used in stock market forecasting. To anticipate stock values, Ghashami et al. (2021) used a hybrid ANN model paired with a particle swarm optimisation approach. The model offers improved the accuracy in comparison to conventional models.

Random Forest, an ensemble learning system, incorporates numerous decision trees, and exhibits stock selection effectiveness. Tan et al. (2019) uses random forest to choose stocks by considering financial ratios and technical indicators. This research confirms the algorithm’s effectiveness in selecting financial assets with high future returns. Furthermore, Long Short-Term Memory (LSTM) Networks, a form of recurrent neural network (RNN), have also been effectively used to record stock price temporal trends. Chung and Shin (2018) introduce an LSTM-based stock market prediction model that generated precise forecasts. Genetic Algorithms

(GAs) are used to optimise trading methods in stock selection. Cheong et al. (2017) develop a GA-based strategy to optimise trading rule selection. This research reveals that the Genetic Algorithms (GAs) can improve the profitability of online trading platforms. HMMs (Hidden Markov Models) have also been used in stock market analysis. HMMs to forecast stock market trends, proving the efficacy of this method in capturing underlying patterns and trends in data (Qiao et al., 2017; Zhang et al., 2019).

Moving average approaches are often used to reduce volatility and discover underlying price trends. Online Moving algorithms, however, have problems such as lagging behind real-time data and being sensitive to window size selection. To solve these constraints, researchers devised online moving mean reversion methods that treat noise and outliers adaptively. Theoretically, Zinkevich (2003) created an online portfolio selection algorithm based on mean reversion, which is a seminal study in using online convex optimisation to stock market prediction. The algorithm is based on the assumption of an unknown market equilibrium price vector that can be approximated using a moving average. The algorithm dynamically allocates the portfolio based on the deviation of the current price vector from the estimated equilibrium price vector, under the premise that prices will eventually revert to their mean. In hindsight, (Zinkevich, 2003) demonstrate that this approach achieves a sub-linear regret bound as compared to the best constant re-balanced portfolio. Further, B. Li and Hoi, (2012), propose Online moving mean reversion algorithm for stock market prediction using relative stock price mean reversion, assuming that underperforming equities would eventually perform well and vice versa. B. Li et al. (2013) proposed algorithms, uses online learning techniques such as confidence weighted learning to update the portfolio vector and models the portfolio vector as a Gaussian distribution. The technique improves forecast accuracy and robustness over single-period mean reversion approaches by adding moving averages to simulate multi-period mean reversion (B. Li and Hoi, 2012).

Subsequent research has improved online moving mean reversion algorithms. (Borodin et al., 2003; B. Li et al., 2013) propose a dynamic threshold for identifying mean reversion signals, updating both the moving average and the threshold using an exponential weighting system that prioritises recent data points. A confidence interval is used to filter out noisy signals and reduce false alarms. In real-world stock data analysis, the algorithm suggested by B. Li et al. (2013) beat baseline strategies such as buy-and-hold, moving average crossover, and constant re-balanced portfolio.

This study proposes a decimated factor online moving mean reversion algorithm for stock market prediction. This algorithm used a combination of decimated data and window size to predict the stock market and stock selection for trading strategies. By employing a hybrid approach that incorporates both decimated data and window size, the algorithm achieved improved prediction accuracy and robustness compared to passive investment approaches. The application of decimated online moving mean reversion algorithm, considering both the decimated factor and window size, holds promise for advancing stock market prediction capabilities. This algorithm offers the potential to capture short-term and long-term dynamics, adapt to changing market conditions, and incorporate diverse factors that influence stock prices. Future research in this area can focus on refining the algorithms, exploring additional factors, and evaluating their performance on more real-world data sets.

3. Research Methodology

This study focuses the Online Moving Average Reversion (OLMAR) and proposes the decimated version called D-OLMAR algorithm. Therefore, we need to briefly describe the formulation of OLMAR and then its extension.

The extant literature provides that most of those formulations that are used in the market conform to the conventional practice of Kelly-based stock selection (Kelly, 1956; Thorp, 1971). To provide a greater level of detail, an investment manager will make a projection for $\bar{X}_{t+1}$ based on k probable values, which include $\bar{X}_{t-1}^{1}, \ldots, \bar{X}_{t+1}^{1}$ in addition to the probabilities that are associated with them $\rho 1, \ldots, \rho k$. It is essential to bear in mind that each $\bar{X}_{t+1}$ represents a unique alternative combination vector of individual price relative estimates. Constructing a portfolio with the objective of getting the best feasible predicted log return is the next step for him or her to take. Considering the limitations discussed by B. Li et al. (2013, 2015) of mean reversion algorithms, named Moving Average Reversion. Wherein, OLMAR assume that $\bar{p}_{t+1} = \bar{p}_{t-1}$ will revert to moving average (MA), where MA_t represent the moving average consistency till the end of period t. In addition, it observes the long-term trend and thus overcomes the drawbacks of existing mean reversion algorithms.

A brief mathematical introduction of Simple Moving Average strategy proposed by B. Li et al. (2015) is as follows:

3.1 Simple Moving Average Reversion

$$\bar{X}_{t+1}(\omega) = \frac{\mathrm{SMA_t}(\omega)}{\rho_t} = \frac{1}{\omega}\left(\frac{\rho_t}{\rho_t} + \frac{\rho_{t-1}}{\rho_t} + \cdots + \frac{\rho_{t-\omega+1}}{\rho_t}\right)$$
$$= \frac{1}{\omega}\left(1 + \frac{1}{X_t} + \ldots\ldots + \frac{1}{\odot_{i=0}^{\omega-2} X_{t-1}}\right) \quad (1)$$

where SMA is Simple Moving Average, ω is the window size, ρ is the price vector and relative change in prices are shown by x and $\odot$ denotes the element-wise product.

OLMAR capture the Creamer (2007a) basic idea of passive aggressive (PA), Creamer (2007b) explain the moving average reversion. PA incorporate the previous output if the classification found correct and adopt aggressively approaches in case of incorrect classification.

3.2 OLMAR Algorithm

The OLMAR optimisation problem presented by B. Li et al. (2015) is given as;

$$\mathrm{b_{t+1}} = \mathrm{arg_{b\in\Delta_m}\min} \frac{1}{2} |\mathrm{b} - \mathrm{b_t}|^2 \quad \boldsymbol{s.t.} \quad b.\bar{x}_{t+1 \geq \epsilon} \tag{2}$$

where $\bar{x}_{t+1}$ is the next price relative which is inversely proportional to last price relative $\bar{x}_t$. In particular, they implicitly assume that next price $\bar{p}_{t+1}$ will revert to last price $\bar{p}_{t-1}$, as follows;

$$\bar{x}_{t+1} = \frac{1}{x_t} \xrightarrow{yields} \frac{\bar{p}_{t+1}}{p_t} = \frac{p_{t-1}}{p_t} \xrightarrow{yields} \bar{p}_{t+1 = \bar{p}_{t-1}} \tag{3}$$

and b_t represents a portfolio vector which is an investment in the market for the $t-th$ periods, i.e $b_t = b_t\,(x_1^{t-1})$ and $b_1^n{}_{= (b_1 \ldots\ldots b_n)}$.

As earlier discussed, the basic idea of this study is the decimation process explored in the of study of de Cheveigné and Nelken (2019) due to challenges faced in the today voluminous trading of high speed and high frequency instruments. The extended version of this model was proposed by Ingber et al. (2020) using multiple and sequential time period to avoid or minimise the human biases errors. Like lowpass filtering a signal and produce some samples from the population (Soleymani and Paquet, 2020). If this process continues without the lowpass filtering concept, it is called down sampling. The core motivation behind this study for decimation is to reduce the cost of processing, time saving and prompt response to the quick opportunities, which create in today fast-moving market. Therefore, the use of a lower sampling rate usually results in a cheaper and faster implementation. We modified the OLMAR algorithm for the decimation process. However, it does not ensure aliasing degree and produces a shift invariant signal representation explored (T. Li et al., 2002).

The term "decimated" refers to the process of resampling the original ("undecimated") data at a lower sampling rate. Therefore, compared to undecimated data, decimated data will have a lower sampling rate. This algorithm, called D-OLMAR, can be stated as:

3.3 Proposed D-OLMAR Algorithm

$$b_{t+1} = arg_{\,b\,\epsilon\,\Delta_m}\ min \frac{1}{2}\ \|b - b_t\|^2 \quad \boldsymbol{s.t.}\ \ \bar{b}\,.y_{t+1} \geq \epsilon \tag{4}$$

where y_{t+1} is the decimated version of x_{t+1} which is $y_{t+1} = x(\,(t+1)M)$ and y_{t+1} is the moving average of $\bar{y}_{t+1}$.

3.4 Assumption

We need the following optimisations for Decimated OLMAR algorithm.

- The given stock data preserve the ergodicity property for a given window size.
- The down sampling factor is chosen such that the first order moment remains constant.

3.5 Proof

The Lagrangian for the optimisation problem of D-OLMAR is:

$$L(b, \lambda, \eta) = \frac{1}{2}\|b - b_t\|^2 + \lambda(\varepsilon - b.\bar{y}_{t+1}) + \eta(b.1 - 1) \quad (5)$$

where $\lambda \geq 0$ and η are the Lagrangian multipliers. Taking the gradient with respect to b and setting it to zero, we get:

$$0 = \frac{\delta L}{\delta b} = (b - bt) - \lambda\bar{y}_{t+1} + \eta_1 \Rightarrow b = b_t + \lambda\bar{y}_{t+1} - \eta_1 \quad (6)$$

Multiplying both sides by 1^T, we get

$$1 = 1 + \bar{y}_{t+1}.1 - \eta_m \Rightarrow \eta = \lambda\bar{y}_{t+1} \quad (7)$$

where $\bar{y}_{t+1}$ denotes the average predicted price relative (market). Plugging the above equation to the update of b, we get the update of b;

$$b = b_t + \lambda(\bar{y}_{t+1} - \bar{y}_{t+1}.1) \quad (8)$$

To solve the Lagrangian multiplier, let us plug the above equation to the Lagrangian;

$$L\lambda = \lambda(\varepsilon - b_t . \bar{y}_{t+1}) - \frac{1}{2}\lambda^2 \|\bar{y}_{t+1} - \bar{y}_{t+1}\|^2 \quad (9)$$

Taking derivative with respect to and setting to zero we get;

$$0 = \frac{\delta L}{\delta \lambda} = \lambda(\varepsilon - b_t . \bar{y}_{t+1}) - \lambda^2 \|\bar{y}_{t+1} - \bar{y}_{t+1}\|^2 \quad (10)$$

$$\Rightarrow \lambda = \frac{\varepsilon - b_t . \bar{y}_{t+1}}{\|\bar{y}_{t+1} - \bar{y}_{t+1}\|^2}$$

Further projecting λ to [0; ∞] we get

$$\lambda = \max\left\{0 \frac{\varepsilon - b_t . \bar{y}_{t+1}}{\|\check{y}_{t+1} - \bar{y}_{t+1}\|^2}\right\} \quad (11)$$

4. Sample size and Frequency

The five rapidly growing stocks of Hi Teck Companies, i.e.: Apple (AAPL), Amazon (AMZN), Alphabet (GOOG), Meta Platforms (META), and Netflix (NFLX) are chosen as sample to test the proposed algorithm. The rationale for selection of these companies is their size, liquidity and presence in the portfolios of institutional investors like mutual funds, pension funds, insurance companies and hedge funds. Additionally, these firms have high weights on indices and are highly monitored by investors. For testing of D-OLMAR, a data of 2739 days' is obtained, starting on May 21, 2012 and ending on October 4, 2023. The window size and frequencies used in the study are mentioned in Table 1.

Table 1. Sampling Criteria

Sample Size	Frequency
10	Daily, Weekly and Monthly
20	Daily, Weekly and Monthly
30	Daily, Weekly and Monthly
40	Daily, Weekly and Monthly
50	Daily, Weekly and Monthly

Source: authors' work.

5. Results

5.1 Data Behaviour

The study of the stock prices of AAPL, AMZN, GOOG, META, and NFLX reveals distinct traits, as shown in the Table 2. NFLX has the highest average price (228.447), followed by META with a mean of 148.658. AAPL has the lowest average price (57.833) Similarly, NFLX has the highest median value (184.210), followed by META with a median of 144.960.

Positive skewness is evident for all companies, showing that the tails of their stock price distributions stretch to higher values. The most skewness (1.013) is found in AAPL, followed by AMZN (0.58), GOOG (0.977), META (0.564), and NFLX (0.575). Kurtosis scores for all companies are negative, signifying smaller tails as compared to a normal distribution. The smallest tails are found in AAPL (-0.533), AMZN (-0.967), META (-0.192), GOOG (-0.057), and NFLX (-0.795). GOOG's distribution is the most typical of the bunch. Line Graphs: Figures from 1 to 5 of each company's stock price fluctuations confirm a flatter distribution than the average one, identifying them as platykurtic.

Table 2. Descriptive Statistics

Parameters	AAPL	AMZN	GOOG	META	NFLX
Mean	62.23	70.867	57.833	148.658	228.447
Median	39.57	54.711	49.609	144.96	184.21
Mode	24.335	11.253	18.014	26.85	51.871
Skewness	1.013	0.58	0.977	0.564	0.575
Kurtosis	-0.533	-0.967	-0.057	-0.192	-0.795

Source: authors' work.

Figure 1. Daily Close Price Movement of AAPL

Source: authors' work.

Figure 2. Price Movements of AMZN **Figure 3. Price Movements of GOOG**

Source: authors' work.

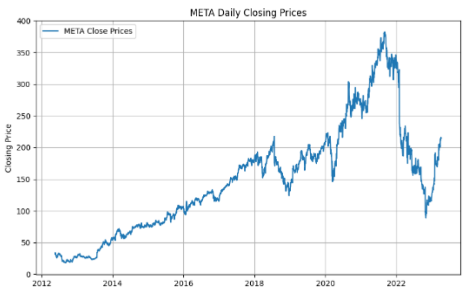

Figure 4. Price Movements of META **Figure 5. Price Movements of NFLX**

Source: authors' work.

6. Simulation and Results

Table 3 provides important information on the mean signal and decimated-mean signal for AAPL, AMZN, GOOG, META, and NFLX for various sample sizes (10, 20, 30, 40, and 50) and frequency (daily, weekly, and monthly). When the values are analysed, notable tendencies appear when the decimated-mean signals are compared to the original mean signals. The decimated-mean signals reveal an increasing tendency across all organisations as the sample size increases and the frequency lowers. For example, when the sample size is increased from 10 to 50 and the frequency is changed from daily to monthly, the decimated-mean signals climb from 0.212 to 0.180. AMZN, GOOG, META, and NFLX all show similar tendencies. These data suggest that the decimation procedure amplifies signals and improves their information content. Thus, the examination of the decimated-mean signals obtained from the table's essential values provides vital insights into the impact of the decimated algorithm on stock price mean signals, demonstrating the relationship between sample size, frequency, and the processed signals.

Table 3. Comparison of Mean with Decimated Mean Signals

Sample Size	Frequency	AAPL		AMZN		GOOG		META		NFLX	
		Mean Signal	D. Mean Signal	Mean Signal	D. Mean Signal	Mean Signal	D. Mean Signal	Mean Signal	D. Mean Signal	Mean Signal	D. Mean Signal
10	D	0.212	0.212	0.200	0.199	0.203	0.208	0.166	0.168	0.177	0.180
10	W	0.212	0.367	0.200	0.382	0.203	0.371	0.166	0.346	0.177	0.398
10	M	0.212	0.679	0.200	0.674	0.203	0.689	0.166	0.708	0.177	0.671
20	D	0.309	0.303	0.308	0.304	0.309	0.309	0.270	0.267	0.280	0.273
20	W	0.309	0.505	0.308	0.495	0.309	0.504	0.270	0.507	0.280	0.596
20	M	0.309	0.991	0.308	0.899	0.309	1.063	0.270	0.894	0.280	0.880
30	D	0.347	0.338	0.372	0.368	0.367	0.356	0.325	0.339	0.341	0.341
30	W	0.347	0.677	0.372	0.614	0.367	0.654	0.325	0.659	0.341	0.675
30	M	0.347	1.309	0.372	1.238	0.367	1.376	0.325	1.033	0.341	1.095
40	D	0.385	0.370	0.418	0.403	0.412	0.392	0.360	0.386	0.411	0.426
40	W	0.385	0.797	0.418	0.731	0.412	0.752	0.360	0.791	0.411	0.724
40	M	0.385	1.526	0.418	1.412	0.412	1.618	0.360	1.176	0.411	1.193
50	D	0.431	0.411	0.461	0.441	0.447	0.422	0.381	0.407	0.492	0.512
50	W	0.431	0.874	0.461	0.837	0.447	0.820	0.381	0.891	0.492	0.767
50	M	0.431	1.793	0.461	1.481	0.447	1.716	0.381	1.229	0.492	1.284

Source: authors' work.

Table 4 shows data on five stocks: AAPL, AMZN, GOOG, META, and NFLX on various window sizes (10, 20, 30, 40, 50) and frequency (daily, weekly, monthly). The figures in the table show the signals of standard deviation as well as the decimated standard deviation for each stock and combination. This can be seen from the data that the decimated standard deviation is slightly smaller than the standard deviation. This implies that the decimated algorithm produces a more precise and accurate representation by removing noise and fluctuation from the signals. Furthermore, as the window size grows, so do the standard deviation and decimated standard deviation values. As a result, a wider window size captures more variance and volatility in stock signals.

Table 4. Comparison of Standard Deviation with Decimated Standard Deviation Signals

Sample Size	Frequency	AAPL		AMZN		GOOG		META		NFLX	
		Std Signal	Decimated-Std Signal	Std Signal	Decimated-Std Signal	Std Signal	Decimated-Std Signal	Std Signal	Decimated-Std Signal	Std Signal	Decimated-Std Signal
10	D	1.209	1.210	1.184	1.185	1.187	1.186	1.183	1.183	1.198	1.199
10	W	1.209	1.185	1.184	1.155	1.187	1.150	1.183	1.177	1.198	1.164
10	M	1.209	1.100	1.184	1.117	1.187	1.083	1.183	1.130	1.198	1.091
20	D	1.311	1.311	1.294	1.294	1.275	1.276	1.273	1.276	1.304	1.304
20	W	1.311	1.237	1.294	1.226	1.275	1.207	1.273	1.261	1.304	1.230
20	M	1.311	1.059	1.294	1.271	1.275	1.094	1.273	1.215	1.304	1.243
30	D	1.345	1.346	1.320	1.322	1.299	1.299	1.324	1.318	1.343	1.338
30	W	1.345	1.252	1.320	1.270	1.299	1.209	1.324	1.267	1.343	1.277
30	M	1.345	0.887	1.320	1.275	1.299	0.908	1.324	1.195	1.343	1.251
40	D	1.350	1.351	1.325	1.326	1.312	1.309	1.347	1.338	1.348	1.348
40	W	1.350	1.225	1.325	1.300	1.312	1.228	1.347	1.303	1.348	1.254
40	M	1.350	0.825	1.325	1.106	1.312	0.811	1.347	1.210	1.348	1.221
50	D	1.350	1.351	1.328	1.330	1.327	1.325	1.357	1.350	1.351	1.349
50	W	1.350	1.206	1.328	1.330	1.327	1.237	1.357	1.320	1.351	1.259
50	M	1.350	0.710	1.328	1.016	1.327	0.791	1.357	1.189	1.351	1.274

Source: authors' work.

For the stocks of AAPL, AMZN, GOOG, META, and NFLX, the Table 5 gives data on the sharp ratio and decimated-sharp ratio for various sample sizes (10, 20, 30, 40, and 50) and frequency (daily, weekly, and monthly). Analysing the values of Table 5 allows us to assess the decimated algorithm's effectiveness in investing decision-making. The sharp ratio is a popular metric for assessing an investment's risk-adjusted performance. A higher sharp ratio suggests a higher risk-adjusted return. We may assess the influence of the decimation process on risk-adjusted performance by comparing the sharp ratio to the decimated-sharp ratio. This study discovered a consistent trend across all sample sizes and frequencies after evaluating the data. The decimated-sharp ratios for each stock roughly match the original sharp ratios. This means that the decimated algorithm effectively retains the equities' risk-adjusted performance. Consider the situation of AAPL, for example. Across different sample sizes and frequencies, the decimated-sharp ratios roughly resemble the original sharp ratios. This pattern also holds true for AMZN, GOOG, META, and NFLX. The decimated algorithm successfully preserves these equities' risk-adjusted performance, demonstrating its usefulness in preserving crucial investment parameters. To make educated judgments, investors rely on precise and reliable performance statistics. The close relationship between the original sharp ratios and their decimated equivalents suggests that the decimated algorithm can be a useful tool in investment decision-making. It ensures that risk-adjusted performance is preserved even after the decimation process, giving investors with reliable data for portfolio optimisation and risk management. As a result of the table values study, we can infer that the decimated algorithm is highly effective in preserving risk-adjusted stock performance, making it a helpful tool for investors in their investing decision-making process.

Table 5. Comparison of Sharp and Decimated Sharp Ratio

Sample Size	Frequency	AAPL		AMZN		GOOG		META		NFLX	
		Sharp Ratio	Decimated-Sharp Ratio	Sharp Ratio	Decimated-Sharp Ratio	Sharp Ratio	Decimated-Sharp Ratio	Sharp Ratio	Decimated-Sharp Ratio	Sharp Ratio	Decimated-Sharp Ratio
10	D	0.175	0.175	0.169	0.168	0.171	0.175	0.141	0.142	0.148	0.150
10	W	0.175	0.310	0.169	0.331	0.171	0.323	0.141	0.294	0.148	0.342
10	M	0.175	0.617	0.169	0.603	0.171	0.637	0.141	0.627	0.148	0.615
20	D	0.236	0.231	0.238	0.235	0.242	0.242	0.212	0.210	0.215	0.209
20	W	0.236	0.409	0.238	0.404	0.242	0.418	0.212	0.402	0.215	0.485
20	M	0.236	0.936	0.238	0.707	0.242	0.972	0.212	0.736	0.215	0.708
30	D	0.258	0.251	0.282	0.278	0.282	0.274	0.245	0.257	0.254	0.255
30	W	0.258	0.540	0.282	0.484	0.282	0.541	0.245	0.520	0.254	0.528
30	M	0.258	1.476	0.282	0.971	0.282	1.515	0.245	0.865	0.254	0.876
40	D	0.285	0.274	0.315	0.304	0.314	0.299	0.267	0.288	0.305	0.316
40	W	0.285	0.650	0.315	0.562	0.314	0.612	0.267	0.607	0.305	0.578
40	M	0.285	1.850	0.315	1.277	0.314	1.994	0.267	0.971	0.305	0.977
50	D	0.319	0.304	0.347	0.331	0.337	0.319	0.281	0.302	0.364	0.379
50	W	0.319	0.725	0.347	0.629	0.337	0.663	0.281	0.675	0.364	0.609
50	M	0.319	2.526	0.347	1.458	0.337	2.171	0.281	1.033	0.364	1.007

Source: authors' work.

Table 6 shows data on the Calmar Ratio and Decimated Calmar Ratio for five stocks: AAPL, AMZN, GOOG, META, and NFLX for various sample sizes (10, 20, 30, 40, 50) and frequencies (daily, weekly, monthly). The Calmar Ratio is a risk-adjusted performance measure that compares the average annual rate of return to the greatest drawdown, offering insight into the risk-reward profile of the investment. The result of table shows that the Decimated Calmar Ratio is higher than the Calmar Ratio for the majority of sample size and frequency combinations. This suggests that the decimated algorithm, which reduces noise and increases signal quality, improves the risk-adjusted performance of investment selections. Higher Decimated Calmar Ratios indicate a better risk-reward trade-off and possibly better investment outcomes. Furthermore, the data explain that the Decimated Calmar Ratios often rise with sample size. This indicates that a larger sample size provides a more robust and reliable estimate of the risk-adjusted performance of the investment. The decimated algorithm contributes to a more accurate assessment of the investment's risk and return potential by capturing a larger amount of previous data. Finally, the result analysis reveals that using the decimated algorithm can improve investment decision-making by enhancing risk-adjusted performance indicators such as the Calmar Ratio. The decimated algorithm delivers more reliable insights into the risk-reward profile of their assets by reducing noise and improving signal quality.

Table 6. Comparison of Calmar and Decimated Calmar Ratio

Sample Size	Frequency	AAPL		AMZN		GOOG		META		NFLX	
		Calmar Ratio	Decimated-Calmar Ratio	Calmar Ratio	Decimated-Calmar Ratio	Calmar Ratio	Decimated-Calmar Ratio	Calmar Ratio	Decimated-Calmar Ratio	Calmar Ratio	Decimated-Calmar Ratio
10	D	0.079	0.079	0.073	0.072	0.077	0.075	0.060	0.060	0.064	0.065
10	W	0.079	0.150	0.073	0.152	0.155	0.075	0.060	0.136	0.064	0.147
10	M	0.079	0.358	0.073	0.279	0.289	0.075	0.060	0.275	0.064	0.331
20	D	0.096	0.094	0.087	0.086	0.088	0.088	0.072	0.071	0.071	0.069
20	W	0.096	0.165	0.087	0.153	0.166	0.088	0.072	0.142	0.071	0.185
20	M	0.096	0.547	0.087	0.284	0.584	0.088	0.072	0.395	0.071	0.296
30	D	0.098	0.095	0.088	0.087	0.105	0.108	0.079	0.083	0.077	0.077
30	W	0.098	0.197	0.088	0.185	0.242	0.108	0.079	0.183	0.077	0.235
30	M	0.098	2.133	0.088	0.506	1.317	0.108	0.079	0.450	0.077	0.416
40	D	0.103	0.099	0.105	0.102	0.103	0.108	0.081	0.087	0.085	0.088
40	W	0.103	0.314	0.105	0.215	0.243	0.108	0.081	0.201	0.085	0.290
40	M	0.103	56.064	0.105	0.829	3.695	0.108	0.081	0.546	0.085	0.546
50	D	0.114	0.109	0.129	0.123	0.109	0.116	0.080	0.086	0.098	0.102
50	W	0.114	0.312	0.129	0.238	0.295	0.116	0.080	0.269	0.098	0.283
50	M	0.114	-3.784	0.129	1.174	16.921	0.116	0.080	0.643	0.098	0.636

Source: authors' work.

7. Discussion

The proposed algorithm, which makes use of decimated data, is useful for improving stock market forecasts and investing choices. The decimated mean, decimated standard deviation, Sharp ratio, and Calmar ratio all points to the fact that with a longer time horizon, returns are higher on average, volatility is lower, and risk-adjusted performance is better. These results demonstrate the benefits of taking a more long-term view when attempting to forecast stock market moves and build portfolios.

The algorithm takes a more objective and rational approach to investment decision-making by using a longer time horizon, hence reducing the likelihood of human error and bias. To further facilitate instantaneous responses to market changes and well-informed judgments, the algorithm makes use of state-of-the-art computational resources to effectively manage and analyse massive amounts of decimated data. Stock market forecasts and stock selection benefit from a longer time horizon, as it allows for the incorporation of underlying patterns, cyclical activity, and mean reversion effects. It helps you see the big picture of market dynamics, so you can choose the best places to put your money and the most promising investments to make.

In addition, sound long-term financial planning techniques are consistent with an expanded perspective of time. In turn, this facilitates more precise forecasting and a methodical approach to investing by providing a more complete picture of fundamental variables, market cycles, and mean reversion events.

Furthermore, the decimated algorithm's efficacy in investment decision-making is demonstrated by its capacity to take advantage of long-term perspectives, to reduce the impact of human biases, to respond swiftly to changing market conditions, and to capture the underlying dynamics of the markets themselves. The algorithm equips investors with the ability to make better-informed decisions, enhance their portfolios, and manage the complexity of the stock market by combining superior computational resources with a holistic understanding of the stock market.

8. Conclusions

This study investigation shows that algorithm-based trading, specifically the D-OLMAR algorithm, has the potential to beat market performance. This strategy has the potential to produce higher trading results by studying massive datasets and making informed judgments. This research discovered that a larger time frame is better suited for predicting and constructing trading strategies since it provides for a broader perspective and the capturing of major trends, hence improving the D-OLMAR algorithm's performance. While algorithm-based trading methods have inherent hazards, our research shows that the D-OLMAR algorithm may effectively decrease long-term risks with suitable risk management measures and continuous monitoring. To maintain the strategy's stability and reliability, it is critical to employ strong risk management measures. The study's findings imply that, as an algorithm-based strategy, the D-OLMAR algorithm has the potential to generate greater risk-adjusted returns while limiting downside risk. The program helps maximise profits while limiting the impact of adverse market situations by leveraging data-driven decision-making processes and applying risk management strategies.

References

[1] Borodin, A., El-Yaniv, R., Gogan, V. (2003), *Can we learn to beat the best stock. Advances in Neural Information Processing Systems*, 16.

[2] Cheong, D., Kim, Y.M., Byun, H.W., Oh, K.J., Kim, T.Y. (2017), *Using genetic algorithm to support clustering-based portfolio optimization by investor information. Applied Soft Computing*, 61, 593-602.

[3] Chung, H., Shin, K. (2018), *Genetic algorithm-optimized long short-term memory network for stock market prediction. Sustainability*, 10(10), 3765.

[4] Creamer, G. (2007a), *Using boosting for automated planning and trading systems*.

[5] Creamer, G. (2007b), *Using boosting for automated planning and trading systems*.

[6] de Cheveigné, A., Nelken, I. (2019), *Filters: when, why, and how (not) to use them. Neuron*, 102(2), 280-293.

[7] Ghashami, F., Kamyar, K., Riazi, S.A. (2021), *Prediction of stock market index using a hybrid technique of artificial neural networks and particle swarm optimization. Applied Economics and Finance*, 8(1), 10-11114.

[8] Gururaj, V., Shriya, V.R., Ashwini, K. (2019), *Stock market prediction using linear regression and support vector machines*. *International Journal of Applied Engineering Research*, 14(8), 1931-1934.

[9] Ingber, L., et al. (2020), *Developing bid-ask probabilities for high-frequency trading. Virtual Economics*, *3*(2), 7-24.

[10] Kelly, W.M. (1956), *Modern Factoring and How it Meets Today's New Financial Requirements. American Bar Association Journal*, 13-17;

[11] Kewat, P., Sharma, R., Singh, U., Itare, R. (2017), *Support vector machines through financial time series forecasting. 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, 2, 471-477.

[12] Li, B., Hoi, S.C.H. (2012), *On-line portfolio selection with moving average reversion. ArXiv Preprint ArXiv:1206.4626*.

[13] Li, B., Hoi, S.C.H. (2014), *Online portfolio selection: A survey*. *ACM Computing Surveys (CSUR)*, 46(3), 1-36.

[14] Li, B., Hoi, S.C.H., Sahoo, D., Liu, Z.-Y. (2015), *Moving average reversion strategy for on-line portfolio selection*. *Artificial Intelligence*, 222, 104-123.

[15] Li, B., Hoi, S.C.H., Zhao, P., Gopalkrishnan, V. (2013), *Confidence weighted mean reversion strategy for online portfolio selection. ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(1), 1-38.

[16] Li, T., Li, Q., Zhu, S., Ogihara, M. (2002), *A Survey on Wavelet Applications in Data Mining. SIGKDD Explor. Newsl.*, 4(2), 49-68, https://doi.org/10.1145/772862.772870.

[17] Qiao, F., Li, P., Zhang, X., Ding, Z., Cheng, J., Wang, H., & others. (2017), *Predicting social unrest events with hidden Markov models using GDELT. Discrete Dynamics in Nature and Society*, *2017*.

[18] Soleymani, F., Paquet, E. (2020), *Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath*. *Expert Systems with Applications*, 156, 113456.

[19] Tan, Z., Yan, Z., Zhu, G. (2019), *Stock selection with random forest: An exploitation of excess return in the Chinese stock market. Heliyon*, 5(8).

[20] Thorp, R. (1971), *Inflation and the financing of economic development*. In *Financing Development in Latin America*, 182-224, Springer.

[21] Zhang, X., Li, Y., Wang, S., Fang, B., Yu, P.S. (2019), *Enhancing stock market prediction with extended coupled hidden Markov model over multi-sourced data. Knowledge and Information Systems*, 61, 1071-1090.

[22] Zinkevich, M. (2003), *Online convex programming and generalized infinitesimal gradient ascent. Proceedings of the 20th International Conference on Machine Learning (Icml-03)*, 928-936.