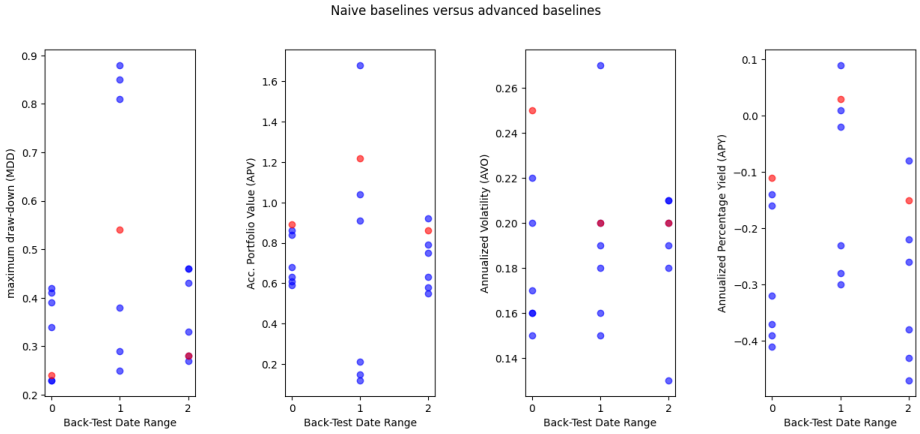

Figure 1: If one compares the performances of existing Portfolio Optimization Strategies (blue datapoints), it turns out there is no significant difference to naive approaches such as buy-and-hold (red datapoints). Datapoints are sourced from the performance table of a recent method [**?**]. From left to right we show MDD, APV, AVO, APY. For MDD and AVO, the lower the better. One tick on the x-axis of each graph corresponds to one of three backtest period. Since we view the credibility of each publication critically, the reported performance of [**?**] itself is excluded and only its measures of baseline performances are compared.

1 Introduction

... Usually, in the scientific development of algorithms that solve a certain task, there is a clear line of improvement over the years of research. However, in our problem, it is unclear how this line of improvement looks like due to a lack of rigid comparison on the same expressive task. Critically, we find evidence that previously proposed methods cannot hold up their claim of superior performance in follow-up work. Figure 1 illustrates that in follow-up work that replicated previous methods, there is no sign of their significance - often the naive approach performs better than many previous methods. The same phenomena can be observed when comparing the measured performances of baselines in [**?**]. ...

2 Problem Derivation

2.1 Preliminaries

2.1.1 Portfolio Optimization Problem

We address the portfolio optimization problem, in which the task is to split the portfolio value on a set of assets A in a manner that maximizes future financial performance metrics π. Specifically, a weight vector $w = \{w_1, w_2, ...w_{|A|}\}, \sum_w = 1$ is to be estimated that assigns the relative size of each position corresponding

to one of the $|A|$ constituents. Because the portfolio should be managed (optimized) continuously, w is to be estimated for every time stamp t_j within the considered time frame $t = \{t_1, t_2, ..., t_{tend}\}$. As a result, the task is to decide on a $|t| \times |A|$ weight matrix W.

2.1.2 Performance Measure

note: the [?] are citations that need to be included. My references throughout the document are basically all the same as in the last slide of the PPT.

The set of assets A is selected to be available for building the portfolio. This might be given through an index such as Hang Seng (HSI) [**?**, **?**] or manually be composed through selecting assets, e.g. Crypto-A in [**?**, **?**].

Previous methods measure their performance on such an asset composition A in a chosen time interval t. We denote this measure as $\pi(A, t)$. For example, if the Cumulative Wealth (CW) of a portfolio built from HSI assets from the beginning to the end of 2023 is 1.13, then $\pi = CW$ and $CW(HSI, 2023) = 1.13$. In financial theory, one can increase the expected return by increasing one's risk. Therefore, measuring return (as in CW) is not sufficient and the reported results need to be complemented by risk- sensitive π such as Sharpe Ratio SR.

Unlike in classification or regression tasks where there is a ground truth which provides true labels for comparison, there is no such ground truth label in the portfolio optimization problem. This is due to the fact that the problem is not formulated as a prediction task, but as an optimization task. Therefore, the aim is to find a strategy or model that maximizes π.

2.2 Statistical Significance of Performance Metrics

2.2.1 Data mining

Research practitioners have a three-dimensional leverage to generate candidate results: Time interval, asset composition and model flexibility. Therefore, given that there are many candidate results, but only few are chosen to report, we must ensure that performance metrics are significant and not a result of leveraging data mining. We first view each dimension of the lever.

Time interval Previous methods select *one* time interval t per asset set A to *report* a performance $\pi(A, t)$. Having a set of candidate time intervals T to *experiment* with, they have $|T|$ possibilities to *select* $t \in T$.

Asset Composition Not only the time interval can be selected, but also the list of available assets A . Because a set (list) A_2 with $a \in A_2, a \in A_1$ reuses asset a from A_1, it is unconditionally correlated with A_1 such that A_1 and A_2 do not represent an independent asset combination. Instead, we are only interested in the possible disjoint sets which are built by combining assets, forming C. Realistically, people are only interested in results on some representative or widely used asset combinations such that researcher practitioners have a constraint on *building* C. In general, we say there are $|C|$ viable possibilities to *select* $A \in C$.

Model Flexibility In theory, deep models can have millions of parameters and changing them yields near infinite possibilities to generate different output

Table 1: The candidate performances $\{\pi\}$ we generated by data mining, $|\{\pi\}| = n_{cand}$. The Σ is a simple additively combined measure of CW,MDD and SR. The indices c, t, f correspond to the tested asset combination, time frame, and strategy, respectively.

	c	t	f	CW	MDD	SR	Σ
Baseline	1	1	-	0.66	-0.52	-0.93	-0.79
π_1	1	1	1	0.62	-0.52	-1.04	-0.94
π_2	1	1	2	0.73	-0.47	-0.72	-0.47
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
Baseline	$\|C\|$	$\|T\|$	$-$	0.40	-0.60	-1.06	-1.26
...	...	...	...	...	...	...	...
$\pi_{n_{cand}}$	$\|C\|$	$\|T\|$	$\|F\|$	0.37	-0.64	-1.04	-1.31

Table 2: The selected performances $\subset \{\pi\}$ we will report. Selection criterion was superior performance relative to baseline. Less surprisingly, all our performances hence look superior over the baseline.

	c	t	f	CW	MDD	SR	Σ
Baseline A	1	2	-	4.78	-0.57	1.70	5.92
Ours A = π_{17}	1	2	9	**6.13**	**-0.53**	**1.89**	**7.48**
Baseline B	3	3	-	15.03	-0.65	2.43	16.81
Ours B = π_{110}	3	3	7	**22.04**	**-0.64**	**2.58**	**23.98**
Baseline C	4	2	-	4.94	-0.41	2.31	6.84
Ours C = π_{119}	4	2	2	**6.81**	**-0.40**	**2.56**	**8.97**

portfolio weights for a given input. In reality, model flexibility is restricted since learning on the training domain will let parameters converge similarly. The model design decisions, such as reward functions, hyperparameters and their fine-tuning strategy on the validation set, are the main sources for *experimenting* with different models F in an attempt to mine $|F|$ candidate results.

Combined there are $n_{cand} = |T| \cdot |C| \cdot |F|$ candidate results for performance measure π. To claim a good performance, $\max(\{\pi_n\}_{n=1}^{n=n_{cand}})$ can then be selected for publication.

2.2.2 Demonstration

It is trivial to generate a superior π. We demonstrate this by randomly generating stategy set F and test each on asset selections C and time frames T, resulting in the performance measure set $\{\pi_n\}_{n=1}^{n=n_{cand}}$.

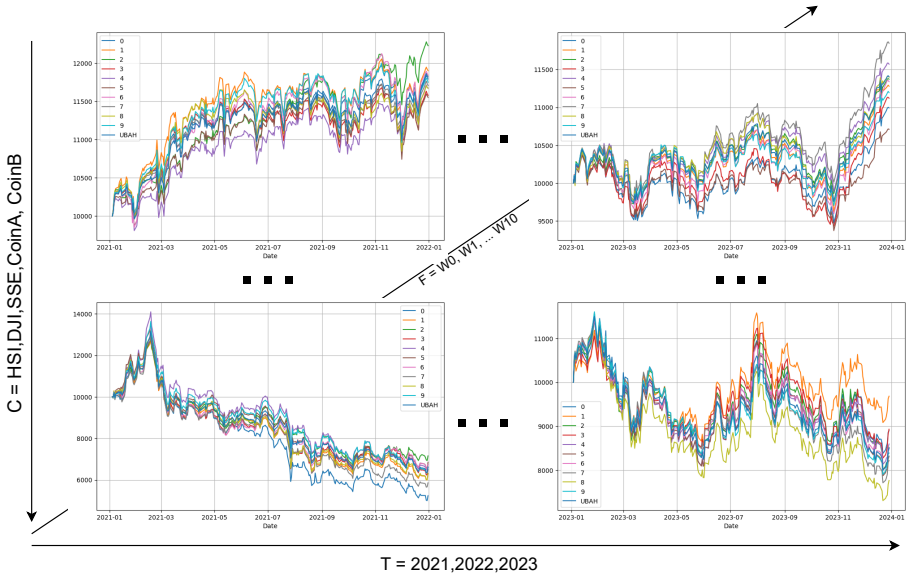

Figure 2: Illustration of the n_{cand} performances for Cummulated Wealth $\cdot 10^4$ we generated over five asset combinations (C), three time frames (T), and 10 random portfolio weight selection strategies (F).

Following [**?**, **?**, **?**, **?**], we measure CW, Maximum Drawdown (MDD) and Sharpe Ratio (SR). For demonstration, we select $|C| = 5$, $|F| = 10$ and $|T| = 3$. We build C from the indices HSI, DJI, SSE, SP500 as well as COIN [**?**, **?**]. Time frames T is one year each, 2021,2022 and 2023. Models F are given through random sampling of weights. Given these, we obtain $n_{cand} = 150$ results as shown in Table 1 and illustrated in Figure 2. Then, we select three π_j with the best relative performance of the baseline to obtain Table 2 for publication: $max_{\pi_j}(\Sigma_j - \Sigma_{Baseline_j}), \Sigma_j = (CW_j + MDD_j + SR_j)$.

2.2.3 Validation using multiple samples

We investigate using a performance distribution instead of a single performance value to tackle the demonstrated problem. Informally, a portfolio optimization strategy would be useless if its performance does not differ from its naive counterparts, e.g. UBAH. To ensure that a strategy is useful, we can test a null hypothesis stating that *"there is no performance difference between learned method M^l and naive method M^n"*:

$$H_0 : \bar{\pi}^{M^l} = \bar{\pi}^{M^n}, \tag{1}$$

where the bar denotes taking the mean.

Previous methods only reported $\pi^{M^l}(A, t)$ on a single asset list A and time interval t. However, with a single sample $\pi_0^{M^l}$ we can make no claims about

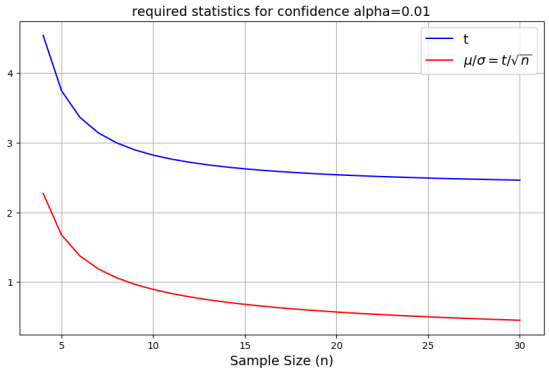

Figure 3: Intuitively, measuring performance on multiple samples increases significance. With higher n, lower performance results are sufficient to support the alternative hypothesis that the learned strategy outperforms the naive strategy. We show the t-statistic in blue and the sample mean over the sample variance in red given the mean performance of the naive strategy is zero.

its significance w.r.t. H_0. To inspect whether or not H_0 can be rejected, we need to measure the performance of method M^l for multiple data samples. In our context, this can be achieved by splitting the time interval in sub-intervals and measuring performance over each. This results in $\pi^{M^l} = \{\pi_{t1}, \pi_{t2}..., \pi_{tn}\}$. Then, we can run a one-sample t-test to determine the probability of observing a value as extreme as $\bar{\pi}^{M^l}$ under the assumption that H_0 is correct. Under H_0,

$$t = \frac{\bar{\pi}^{M^l} - \bar{\pi}^{M^n}}{std(\pi^{M^l})/\sqrt{n}} \tag{2}$$

follows a t-distribution. Therefore, the probability to observe a mean greater or equal (g.eq.) $\bar{\pi}^{M^l}$ under H_0 is

$$p(tvar \geq t|H_0) = 1 - CDF_t(t, n-1), \tag{3}$$

where $tvar$ is a t-distributed random variable and CDF_t the cummulative t-distribution function.

Increasing n decreases the probability to observe a mean g.eq. $\bar{\pi}^{M^l}$. Reporting results on more samples is hence required to let the performance measure be significant. To reach a confidence interval of α, we need the p value from eq. (3) to be smaller than α. Figure 3 shows the critical t-value required for $\alpha = 0.01$ as a function of n.

So far we inspected the performance distributions which are required for significant results *given one sampling attempt*. Now, given the possibility of data-mining as described in section 2.2.1, we must consider n_{cand} to avoid inflating Type-I error simply due to p-hacking. The critical value can be adjusted using Bonferroni Correction [**?**] to α/n_{cand}. Given the $n_{cand} = 150$ from our demonstration, the p-value in eq. (3) must become as small as 0.000067.

Table 3: Lowest p-values observed per metric when measuring performance monthly (n=12) to collect a sample distribution per $\pi(c, t, f)$. The table shows 3 out of n_{cand} candidate performances that had the minimal p-value w.r.t. H_0. None of the results is significant ($p < \alpha$), in line with what we would expect for a strategy that assigns portfolio weights randomly.

	c	t	f	p(CW)	p(MDD)	p(SR)
$\min_\pi p(CW_\pi)$	1	0	0	**0.34**	0.60	0.39
$\min_\pi p(MDD_\pi)$	1	2	8	0.40	**0.14**	0.50
$\min_\pi p(SR_\pi)$	2	0	6	0.34	0.23	**0.31**

Now, if we want to run such test with real data as in Table 1, we need to ensure that the preconditions for the test are met. One precondition is that the sample means are normally distributed, which is, even if the population is not normal distributed, true for large n due to the Central Limit Theorem. Howver, because we want to keep our derivation valid for small sample sizes, we should transform the metrics so that under H_0 they are normal distributed. Since returns ($price_{t_1}/price_{t_0}$) are log-normal distributed, we can take the log of the prices or returns to get them and consequently also the metrics normally distributed under H_0.

We reuse our generated results from Table 1, but now a) consider log-prices and b) split the 1-year time interval to get monthly results, as such yielding a distribution for each $\pi(A, t, f)$ with mean and standard deviation; $n = 12$ and t is one year long. Each thus obtained sample distribution we plug in eq. (2) to obtain the t-statistic and its p-value from eq. (3). Then we can compare it with α to argue about π's significance.

Table 3 shows the results, which indicate that by the proposed *measuring performance distribution* instead of *measuring a single performance*, the demonstrated *"it is trivial to generate a superior π"* no longer holds, as such the problem is effectively solved by redefining *superior*.

3 Method

3.1 Unified Benchmark

We propose a benchmark with a selection of assets and time frames so that previous work and future work can be fairly compared with each other. Our unified benchmark effectively removes the data mining leverage in the c and t dimensions that we have discussed in section 2.2.1 and thus increases credibility. Moreover, we propose adjusted metrics that take into account the insights from section 2.2, as such solving current evaluation problems.

Dataset We select $|T| = 3$ time frames and $|C| = 5$ asset combinations to build a fixed benchmark for the evaluation of Portfolio Optimization Strategies.

Table 4: The proposed dataset for benchmarking.

Dataset	$\|A\|$	Test Range 1 2020	Test Range 2 2023	Test Range 3 1990
HSI	30	✓	✓	no easy access
DJI	30	✓	✓	✓
SNP	500	✓	✓	✓
CoinA	10	✓	✓	not exist
CoinB	50	✓	✓	not exist

Unlike previous work, we require a method not only to be compared on j index-time pairs ($A_j \in C, t_j \in T$), commonly $j = 3[\textbf{?}, \textbf{?}, \textbf{?}]$, but instead on all possible pairs $(A, t) \in (C \times T)$. Table 4 shows the dataset composition. The considered criteria for selection are:

- Easily accessible price histories (yahoo finance)
- Covering sufficiently long test period
- Covering different asset classes (stocks, currencies)
- Covering bull and bear markets
- Covering different asset pool size (i.e. $|A|$)
- Covering at least one time frame before automated AI based trading was practiced in the industry

HSI resembles the China focused Hang Seng Index with 30 assets. DJI is the Dow Jones Index, containing the 30 largest US companies. SNP is the US SNP500 index with 500 assets. Coin A is a cryptocurrency datasets covering the 10 symbols with the highest market capitalization. Coin B covers the next 50 most valuable cryptocurrencies coming after Coin A.

We do *not* specify a training domain as developers should enjoy flexibility in their innovations. Given that there are methods which use not only historical data, but also news data or operating numbers, the benchmark should be open towards different approaches that try to leverage existing data. Access to *any* future data in the test period however must remain strictly forbidden.

Metrics As illustrated in table 5, previous work [1,2,3,4,5,6,7] used a variety of evaluation metrics even though they attempted to solve the same task. To resolve this inconsistency, for our benchmark the metrics which are most characteristic for the portfolio optimization is selected: APY, AVO and MDD.

Given the assumption of a drift in the underlying stochastic process, the expected value of Cummulated Wealth (CW) is not constant over time. Specifically, the longer the time interval observed, the more the drift accumulates. CW can be fully expressed as a monotonically increasing function of Annualized Percentage Yield (APY) which is time-normalized, so that there is no need

Table 5: The metrics existing across previous works and those selected for the proposed benchmark.

Acronym	Full Name	Selected
CW	Cummulated Wealth	**X**
APY	Annualzied Percentage Yield	✓
AVO	Annualized Volatility	✓
SR	Sharpe Ratio	**X**
MDD	Maximum Drawdown	✓
CR	Calmar Ratio	**X**
SOR	Sortino Ratio	**X**

to report CW results: $CW(A,t) = (APY(A,t)+1)^y$, where y is the covered interval t expressed in number of years.

Similarly, the Sharpe Ratio can be derived given APY and AVO, such that it provides no further information content. As a consequence, for the proposed benchmark only APY, AVO and MDD are relevant.

Moreover, as we derived in section 2.2, future authors are encouraged to obtain a performance distribution over multiple samples and subsequently do significance testing against the baselines. To this end it is possible to split each of our time frames from Table 4 in n subintervals and obtain the n-sized sample set of proposed as well as baseline (or naive) performance. Following eq. (1)-eq. (3), significance can be reported.

3.2 Market versus Random Walk Discrimination Test

The proposed unified benchmarks fixes the asset combination and time frame leverage (A,t), but experimenting with multiple models and strategies (f) until one yields superior results is still possible. We attempt to address this by comparing a method's performance on simulated random walks.

In accordance with section 2.1.1, DRL Portfolio optimization strategies and machine learning attempts in general try to learn a mapping from inputs (historical prices, optionally also news or operating nubmers) to outputs (weight allocation). If the inputs were purely random, then there is no pattern to learn. As a consequence, the output leading to maximum expected $\bar{\pi}$ can be determined without looking at the data, instead derived from the characteristics of the random distribution. Any method and metric that also yields good results on randomly generated simulated inputs lacks credibility as we cannot determine whether it could make use of market data. For a price series, dynamics are modelled by Geometric Brownian Motion (GBM) [**?**]. We a) set up a procedure to verify whether a portfolio optimization method performs any different on a randomly generated simulated prices series versus on real market data. We b) then also show how usage of our benchmark's metrics intrinsically mitigates this issue. For a), for each asset in asset pool A, we replace its real price by

simulated GBM

$$p_t = p_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right), \tag{4}$$

where W_t is the Brownian Motion and μ, σ is set to match the empiric market distribution. Having thus a price series $p_0, ..., p_{t_{end}}$ for each asset as in the real setting, the portfolio optimization strategy can choose allocation weights, the evaluation protocol remains also the same.

It is obsolete to show again that also on simulated prices there will be a superior performance given large enough candidate performances as discussed in section 2.2.1. For b), with the proposed unified benchmark we have already greatly reduced data mining opportunities. More interesting, however, we can compare the performance distributions on simulated versus market data, drawing on the scheme from section 2.2, thus obtaining significance measures for performance on both market data (Table 7) and simulated data (Table 8). On a generated GBM where we know its distribution and pure randomness, no strategy should be able to generate significant excess returns, hence the naive baseline should perform similarly, whereas on the real market data hidden patterns or exploitable distributions are hypothetically possible. Therefore, if a method proofs to be significant on the market, but insignificant on the GBM, this is evidence for a truly effective method. In contrast, if it yields significance on both market and GBM, it should be interpreted with care. Rather than showing an effective method, this case would be an indicator that the significance testing method itself might has a methodical issue, for instance because of violated preconditions. In this case, one needed to analyze the specific statistical relationship between the variables and adjust the testing procedure accordingly.

4 Experiments

To demonstrate the effectiveness of our approach, we contrast previous evaluation protocols with ours. We show how previous attempts could yield insignificant but apparently superior results on real market data and how our method fixes this problem.

We evaluate two naive baselines (UBAH, Market) and two rule-based strategies on our benchmark. Table 6 shows the results.

The main idea of strategy 1 is to pay attention to the price changes of assets and adjust the its weight based on these changes. We name it Dynamic Price Feedback Strategy (DPFSStrategy). Specifically, on the first day of investment, funds are allocated in equal amounts to each asset. From the next day, the strategy will pay attention to the changes in each asset. If the price of an asset increases compared with the previous day, the weight of the asset will be reduced; if the price decreases, the investment in the asset will be increased. This is largely based on contrarian investing thinking: if the price of an asset rises significantly, it may become overvalued, thereby increasing the risk of a future decline. Conversely, if an asset's price drops significantly, it may be

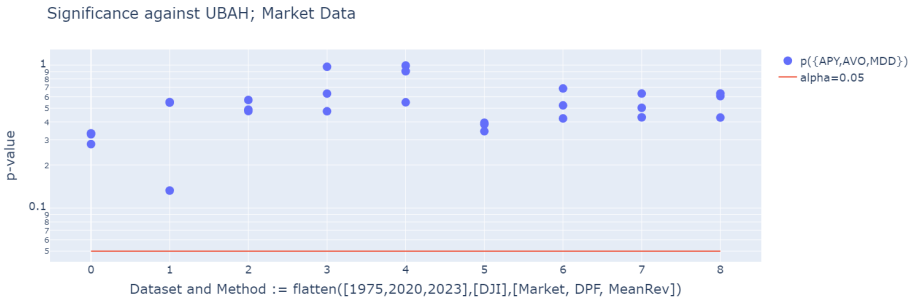

Figure 4: Significance of performance on market data visualized, from Table 7. As expected, no method exhibits significance (red horizontal line) against the naive UBAH baseline, measured on several time frames.

undervalued and thus likely to rise in the future. This feedback mechanism is designed to capitalize on price fluctuations for investing. After the weight adjustment is completed, buy and sell operations are performed based on the asset allocation weight.

Strategy 2 is based on the assumption that asset prices will fluctuate around their historical average prices and eventually converge to this average. When the asset price deviates from the historical average, the strategy will regard it as an opportunity to buy or sell. We name it the Mean Reversion Strategy (MRStrategy). In detail, funds are allocated equally to each asset on the first day of investing. Starting from the next day, the average price of a certain time window in the past is calculated for each asset. If the size of time window is greater than the number of days with known historical prices, no buying or selling occurs. When the number of days with known historical prices is equal to the size of the time window, the following judgment is made: if the current price of the asset is lower than its historical average price, increase the weight of the asset (bullish); if the current price is higher than the historical average price, decrease the weight (bearish).). After the weight adjustment is completed, buy and sell operations are performed based on the asset allocation weight.

We also conduct significance testing based on section 2.2 and show the results in Table 7. Moreover, this significance can be compared with the significance when replacing the benchmark by a random simulated price series, as reported in Table 8. note: I added the following sentence:From the results we can observe the expected behaviour, that, there is no easy way to generate significantly superior results on all three APY, AVO, MDD metrics using simple methods, whereas on the single value results from Table 6 it appears as if some methods were more suitable. note: end of sentence.

Previous work did not publish their source code. Reimplementing them is not part of this work. Comparing their performance on the proposed benchmark is the next logical step for future work. Only then we can understand the state of the art on the task.

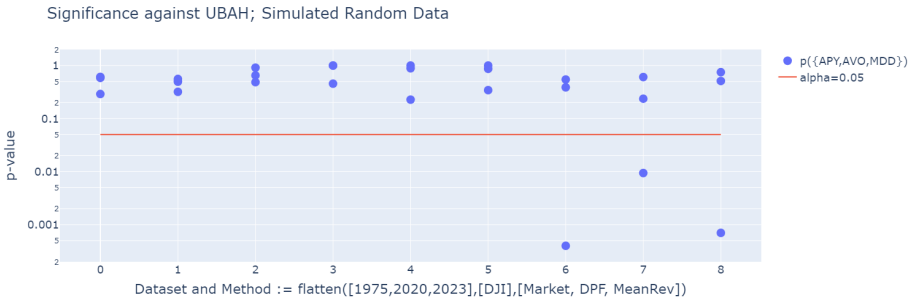

Figure 5: Significance of performance on simulated data visualized, from Table 8. Here we can see that even though there are three samples below α, it is only one metric each, showing that good performance on one metric still comes at the cost of another, e.g. gain returns at the cost of risk. Interestingly, if one compares with Table 8, it is the *also naive* Market Capitalization strategy which shows this significantly low MDD values against UBAH.

Table 6: Results on our new benchmark.

		HSI		DJI			SNP500			COIN A		COIN B	
Metric	Algo	2020	2023	2020	2023	1975	2020	2023	1975	2020	2023	2020	2023
APY	UBAH	0.86	-0.15	0.07	0.14	0.48	0.15	0.16	0.39	2.09	**1.74**	**9.50**	**2.32**
	Market	0.74	-0.11	**0.31**	**0.33**	**0.52**	**0.39**	**0.43**	0.40	**3.07**	1.42	4.86	1.77
	DPF	0.81	**-0.07**	1e-6	0.12	**0.42**	0.09	0.16	0.05	1.57	1.17	7.47	1.55
	MeanRev	**1.11**	**-0.07**	0.18	0.12	0.50	0.36	0.18	0.42	0.90	1.10	6.01	1.94
AVO	UBAH	0.43	**0.31**	**0.35**	**0.12**	0.20	**0.36**	**0.14**	0.15	0.57	0.40	0.96	0.47
	Market	**0.41**	0.33	0.37	0.14	0.20	**0.36**	0.16	0.17	0.59	0.35	0.78	0.43
	DPF	0.43	0.32	0.37	**0.12**	0.20	0.41	**0.14**	**0.04**	0.53	**0.32**	0.66	0.40
	MeanRev	0.45	0.33	0.37	0.13	**0.19**	**0.40**	0.15	0.14	**0.51**	0.35	**0.62**	0.41
MDD	UBAH	0.24	0.30	0.33	**0.09**	0.17	0.38	0.13	0.16	0.49	0.27	0.54	0.37
	Market	0.25	0.29	**0.30**	**0.09**	**0.16**	**0.33**	**0.10**	0.16	0.52	**0.21**	0.54	0.33
	DPF	0.25	0.28	0.38	0.10	0.18	0.43	0.13	**0.14**	**0.46**	**0.21**	**0.47**	0.34
	MeanRev	**0.23**	**0.26**	0.34	0.10	0.17	0.40	0.13	0.15	0.49	0.22	0.52	**0.31**

Table 7: p-values with respect to naive baseline UBAH when sampling metrics monthly.

		HSI		DJI			SNP500			COIN A		COIN B	
Metric	Algo	2020	2023	2020	2023	1975	2020	2023	1975	2020	2023	2020	2023
APY	Market	**0.61**	**0.45**	0.28	0.13	0.48	0.34	0.12	0.46	0.36	0.67	0.84	**0.66**
	DPF	0.53	0.42	**0.33**	**0.55**	**0.57**	**0.52**	**0.50**	**0.63**	0.68	**0.75**	0.95	0.70
	MeanRev	0.29	0.44	**0.33**	**0.55**	0.49	0.32	0.47	0.45	**0.76**	**0.75**	**0.97**	0.61
AVO	Market	**0.26**	0.85	**0.48**	0.99	0.40	**0.56**	0.94	0.93	0.50	0.04	0.02	0.16
	DPF	0.54	**0.78**	0.97	**0.55**	0.39	0.67	**0.58**	**0.27**	0.26	**0.01**	**1e-6**	**0.04**
	MeanRev	0.71	0.80	0.63	0.91	**0.34**	0.69	0.79	0.28	**0.17**	0.10	**1e-6**	0.06
MDD	Market	0.47	0.39	0.69	0.63	**0.43**	0.57	0.68	**0.26**	0.60	0.97	**0.82**	**0.81**
	DPF	0.44	**0.38**	**0.42**	0.50	0.61	**0.40**	0.54	0.59	**0.73**	**0.96**	0.99	0.83
	MeanRev	**0.41**	0.44	0.52	**0.43**	0.63	0.43	**0.42**	0.64	0.74	**0.96**	0.99	0.91

Table 8: p-values with respect to naive baseline UBAH on simulated random price movements instead of real market data. Comparable with Table 7.

		RW 1		RW 2			RW 3			RW 4		RW 5	
Metric	Algo	2020	2023	2020	2023	1975	2020	2023	1975	2020	2023	2020	2023
APY	Market	0.35	0.15	0.58	0.32	**0.91**	0.38	**0.96**	0.23	**0.57**	0.58	0.48	0.45
	DPF	0.42	0.30	0.29	0.50	0.65	0.42	0.51	0.47	0.31	0.43	0.23	**0.60**
	MeanRev	**0.43**	**0.50**	**0.61**	**0.56**	0.48	**0.46**	0.34	**0.57**	0.54	**0.67**	**0.80**	0.42
AVO	Market	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	DPF	0.75	0.97	1.00	0.89	0.87	0.76	0.75	0.99	0.89	0.87	0.91	0.96
	MeanRev	**0.33**	**0.41**	**0.45**	**0.23**	**0.34**	**0.32**	**0.26**	**0.31**	**0.13**	**0.07**	**0.31**	**0.19**
MDD	Market	**1e-3**	**1e-2**	**4e-4**	**9e-3**	**7e-4**	**1e-6**	**2e-5**	**4e-4**	**4e-3**	**0.10**	**1e-6**	**1e-6**
	DPF	0.38	0.36	0.39	0.24	0.51	0.40	0.36	0.53	0.45	0.25	0.52	0.33
	MeanRev	0.69	0.61	0.54	0.61	0.75	0.65	0.68	0.60	0.57	0.53	0.28	0.67