Statistical Overfitting and Backtest Performance

David H. Bailey[1], Stephanie Ger[2], Marcos Lopez de Prado[3], Alexander Sim[4], Kesheng Wu[5]

Abstract

In the field of mathematical finance, a "backtest" is the usage of historical market data to assess the performance of a proposed trading strategy. It is a relatively simple matter for a present-day computer system to explore thousands, millions or even billions of variations of a proposed strategy, and pick the best performing variant as the "optimal" strategy "in sample" (i.e., on the input dataset). Unfortunately, such an "optimal" strategy often performs very poorly "out of sample" (i.e., on another dataset), because the parameters of the investment strategy have been overfit to the in-sample data, a situation known as "backtest overfitting" (Bailey and Lopez de Prado [2012, 2014], Bailey et al. [2014, 2015]).

While the mathematics of backtest overfitting has been examined in several recent theoretical studies, here we pursue a more tangible analysis of this problem, in the form of an online simulator tool. Given a input random walk time series, the tool develops an "optimal" variant of a simple strategy by exhaustively exploring all integer parameter values among a handful of parameters. That "optimal" strategy is overfit, since by definition a random walk is unpredictable. Then the tool tests the resulting "optimal" strategy on a second random walk time series. In most runs using our online tool, the "optimal" strategy derived from the first time series performs poorly on the second time series, demonstrating how hard it is *not* to overfit a backtest. We offer this online tool to facilitate further research in this area.

1. Introduction

Modern high-performance computing technology, accelerated by the relentless advance of Moore's Law, has enabled researchers in many fields to perform computations that would have been unthinkable in earlier eras. For example, in the July 2014 edition of the Top 500 list of the world's most powerful supercomputers (see Figure 1), the best system performs at over 30 Pflop/s (i.e., 30 "petaflops" or 30 quadrillion floating-point operations per second), a level that exceeds the sum of the top 500 performance figures approximately ten years earlier (Simon [2015]). Note also that a 2014-era Apple MacPro workstation, which features approximately 7 Tflop/s (i.e., 7 "teraflops" or 7 trillion floating-point operations per second) peak performance, is roughly on a par with the #1 system of the Top 500 list from 15 years earlier (assuming that the MacPro's Linpack performance is at least 15% of its peak performance).

[1] Lawrence Berkeley National Laboratory (retired), and University of California, Davis, Department of Computer Science.

[2] Boston College, Department of Mathematics, and Lawrence Berkeley National Laboratory, Science Undergraduate Laboratory Internship (SULI) program.

[3] Senior Managing Director at Guggenheim Partners, New York, and Research Affiliate at Lawrence Berkeley National Laboratory.

[4] Lawrence Berkeley National Laboratory, Scientific Data Management (SDM) group.

[5] Lawrence Berkeley National Laboratory, Scientific Data Management (SDM) group.

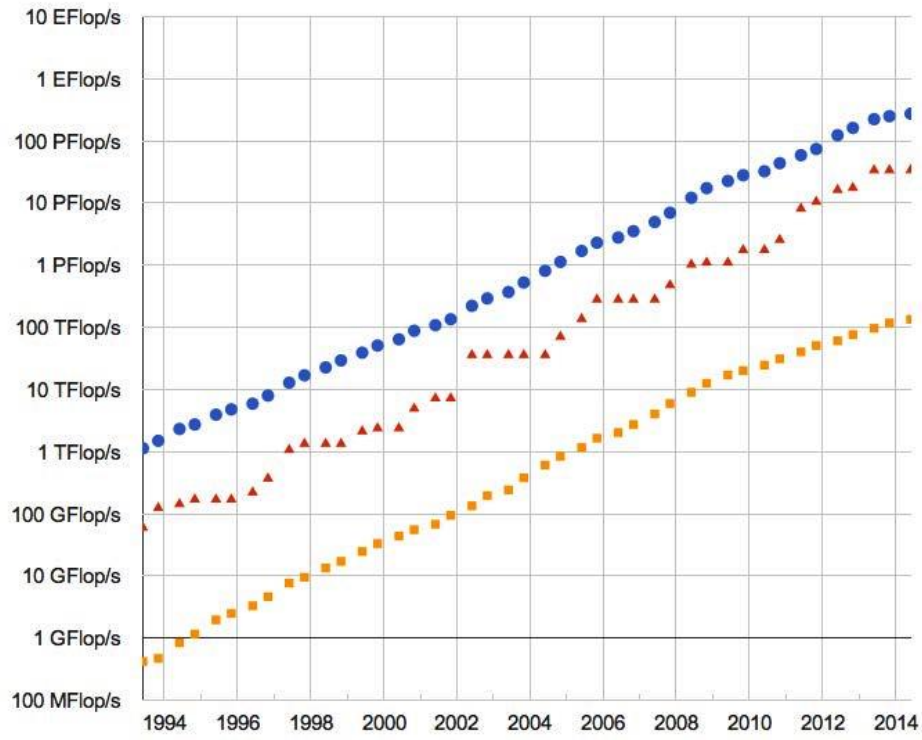


Figure 1: Performance of the Top 500 computers: Red = #1 system; orange = #500 system; blue = sum of #1 through #500.

These powerful computer systems make it possible to analyze very large datasets and also to simulate very complex natural phenomena. However, they also permit researchers to explore thousands, millions or even billions of variations of a proposed model on a given dataset, and thus greatly magnify the potential for statistical overfitting errors.

Statistical overfitting, in this context, means either proposing a model for an input dataset that inherently possesses a higher level of complexity than that of the input dataset being used to generate or test it, or else trying many variations of a model on an input dataset and then only presenting results from the one model variation that appears to best fit the data. In many such cases, the model fits the data well only by fluke, since it is really fitting only the idiosyncrasies of the specific dataset in question, and has little or no descriptive or predictive power beyond the particular dataset used in the analysis.

Statistical overfitting can be thought of as an instance of the "selection bias," wherein one presents the results of only those tests that support well one's hypothesis. These types of errors are discussed in David Hand's very readable 2014 book *The Improbability Principle* (Hand [2014]).

Statistical overfitting and "selection bias" are thought to be at the root of some reproducibility problems that have plagued several fields of scientific research in recent years. For example, in the biomedical field, there have been numerous instances of pharmaceutical products that look

promising based on initial clinical tests and trials, but later disappoint in real-world implementation. The success rates for new drug development projects in Phase II trials have recently dropped from 28% to 18% (Prinz et al. [2011]). The principal reason for these disappointments is now thought to be the fact that pharmaceutical firms, intentionally or not, typically only publish the results of successful trials, thus introducing a fundamental bias into the results.

Recently the U.S. Securities and Exchange Commission announced that its examination of hedge funds uncovered a number of issues that are examples of "selection bias." According to a *Wall Street Journal* report, the SEC "uncovered marketing and advertising issues, with some firms potentially misleading clients on past performance by 'cherry picking' their results from fund to fund" (Ackerman [2014]).

2. Backtest overfitting in finance and investments

In the field of mathematical finance, statistical overfitting most often arises when using "backtests" to develop and/or refine an investment strategy, a phenomenon known as "backtest overfitting." The term "backtest" means using historical market data (e.g., the past ten years of daily S&P500 closing averages) to evaluate how a proposed strategy would have performed had it been fielded over the past time frame in question. Since even a modestly powerful desktop or workstation can explore thousands, millions or even billions of variations of a strategy, it is not only possible but in fact quite likely that some variation of the strategy will perform well on this backtest dataset, yet in reality not have any useful predictive power, since the proposed strategy is only fitting idiosyncrasies in the "noise" of the dataset. Overfitting can also occur when a statistical test is carried out multiple times on the same dataset, without controlling for the steady increase in the false positive rate (this is known as the "multiple-testing problem").

As an example, if someone rolls a set of ten six-sided dice, the probability of seeing all sixes (or any other particular pre-specified combination) is approximately 1.65×10^{-8}, or in other words, roughly one chance in 60 million. But as she rolls the ten dice together over and over again, her chances of getting all sixes increase, until, after tens of millions of trials, she is almost guaranteed to see a roll with ten sixes. If one had seen only the one all-sixes result of this experiment, one might have been justified in concluding that the dice are "loaded," and that future rolls are likely to produce disproportionate numbers of sixes, but this is not the case.

Rolling ten dice 60,000,000 times is perhaps not a practical real-world scenario. But using a computer to explore 60,000,000 variations of an investment strategy is a relatively minor task, something that could be done in a few minutes on a present-day system. Hence, such computer "experiments" are vastly more likely to result in overfitting errors.

As another illustrative example, suppose that a financial advisor sends out 10,240 ($= 10 \times 2^{10}$) letters to prospective clients, with half predicting that some stock or other security would go up in market value, and half predicting that it would go down. One month later, the advisor sends out a set of 5,120 letters, only to those who were earlier sent the correct prediction, again with half predicting some security will go up and half predicting it will go down. After ten repetitions of this process, the final ten recipients, were they not aware of the many letters to other clients, doubtless would be impressed at the advisor's remarkable prescience. The set of ten correct

predictions sent to each of these final ten recipients can be thought of as the equivalent of the string of ten consecutive sixes in the first example above.

As a third example, suppose that an investor believes that there are daily, weekly or monthly patterns in historical stock market data, and he or she seeks a strategy that can exploit these patterns for financial gain. One very basic strategy would be to buy a set of stocks each Monday and then sell them on Wednesday. Another would be to buy stocks on the sixth day of each month and sell them on the 19th. A computer program can easily explore many thousands of such variations. The strategy could then be refined further by selling the portfolio at any time that it drops in value more than 10% from its initial price, or to purchase shares only when they increase in value by 10% over their value at the start of the trading period, as part of a strategy to capture "momentum" in market prices. There are enormous numbers of such combinations – millions just for this simple example – which are the equivalent to rolling the dice many times in the first example above. And, for the same reason, it is highly likely that one of these parameter combinations will perform well on the historical dataset, but this is merely a "selection bias" statistical fluke.

Harvey and Liu [2014] and Harvey et al. [2014] report hundreds of examples where multiple testing and selection bias have taken place in the factor investing literature. The list is by no means exhaustive. In fact it is very difficult to find publications where multiple testing has been controlled for when discovering new factors. This leads these authors to conclude that *"most claimed research findings in financial economics are likely false"*. Bailey and Lopez de Prado [2014], Bailey et al. [2015] and Harvey and Liu [2015a, 2015b] have proposed practical procedures to correct for the increased false positive probability that results from multiple testing.

Backtested models are usually based on a hypothetical phenomenon governing financial markets. However, as some have noted, "poorly performing strategies are discarded or optimized to create the final product" (Beaudan [2013]), thus unwittingly introducing a bias into the analysis. Indeed, it now appears that backtest overfitting errors are much more pervasive in the field than commonly recognized, and likely are a principal reason that many systematic funds, which strategies rely on backtests, often disappoint (Bailey et al. [2014]). Such errors evidently are an unfortunate byproduct of fast, computer-based tools used by analysts to explore, develop and refine potential models and strategies.

3. Quantifying backtest overfitting effects

How can one quantify the effects of backtest overfitting? A common statistic utilized to measure performance is the Sharpe ratio, as it can be "used to quantify the backtested strategy's return on risk" (Lopez de Prado [2013]). The Sharpe ratio, informally speaking, is defined as the performance of an investment over a given period, normalized by the standard deviation of the investment's changing value over that period. For a more precise definition and other technical details, see Lopez de Prado [2013] or Bailey et al. [2014].

While the Sharpe ratio on the backtest dataset is important, one must also consider the Sharpe ratio for the algorithm on new data. In an attempt to avoid backtest overfitting, researchers and analysts often use the "hold-out" method to a strategy, which consists in splitting the data into

two subsets. The model or strategy is trained in one subset (called the In Sample, or IS dataset) and tested on another subset (called the Out of Sample, or OOS dataset). While this form of cross-validation is useful for some purposes, unfortunately it is not a guarantee against overfitting, since the hold-out method does not control for the number of trials involved in a discovery. And, if one tries hard enough, one can find an "optimal" strategy that performs well on both the In Sample and Out of Sample datasets, yet still has no substantive "skill."

Two of the present authors, together with two other colleagues, co-authored some recent studies on backtest overfitting. In the first study (Bailey et al. [2014]), a formula was derived relating the number of variations attempted in the development of a strategy to the size of the backtest dataset. For example, it was shown that if only five years of daily market data are available, and if 45 or more independent variations of a strategy are tried, it is more than likely that the best strategy selected in this process has a Sharpe ratio of 1.0 or better, indicating one standard deviation above the mean in performance. In the second study (Bailey et al. [2015]), a formula was derived for the probability of backtest overfitting. Numerous other results are presented in both papers. One particularly troubling consequence of this theory is that overfitting a backtest on time series with memory (e.g., autoregressive processes) leads to persistent losses, rather than just zero expected performance.

While a theoretically rigorous basis for backtest overfitting is important for fundamental understanding, it is clear that some additional research tools are needed. For example, as mentioned above, some researchers and analysts believe that markets may act cyclically or seasonally. While that hypothesis is likely to be true in some specific cases, a carefully designed experiment is required to reach any conclusion with a satisfactory degree of confidence. Preliminary studies have shown that in most cases, there simply are not enough data points to determine the statistical significance of these cyclical behaviors, after controlling for the number of trials involved in making those supposed discoveries. Several other hypotheses of this sort could be listed. Which of these ideas have merit and which do not?

4. An online demonstration of backtest overfitting

In this work, we present an online tool that allows analysts and researchers to experiment with the phenomenon of backtest overfitting. We have based the tool on a popular investment strategy, with a very limited number of possible parameter choices, in part to emphasize the fact that backtest overfitting can arise even in simple contexts, and is not only a potential problem for highly sophisticated strategies. For our test data, we simulate a series of daily prices by drawing returns from a Gaussian (normal) distribution, using a high-quality pseudorandom number generator.

The current version of the tool is available online at this URL:
http://datagrid.lbl.gov/backtest

4.1. Simple Example of Backtest Overfitting (SEBO)

The investment strategy considered in our tool is particularly simple. We assume an investment decision is being made monthly, and only one equity with the simulated price is considered. A day of the month is chosen as the entry day for the investment. The strategy enters the market on either the buy side (long) or the sell side (short), and it always commits all available funds to this

single position. The strategy exits the market either after it has held the equity for the specified number of days or once it triggers a stop loss condition. The variables controlling this simple strategy are then adjusted to produce good results, measured by the Sharpe ratio, based on the input time series dataset.

Our implementation of this web site consists of text introducing the parameters that control the process. A Python program then accepts input values from users, performs the computation and displays the output results. For convenience of discussion, we have named the program "Simple Example of Backtest Overfitting" (SEBO). The following is a detailed description of its operation:

1. SEBO first constructs a time series simulating stock market data. The daily price fluctuations are simulated by drawing returns from a standard Gaussian (normal) distribution, which are then compounded to derive a price time series. We use the pseudorandom number generator *random.gauss* from the Python programming language. The simulated prices generated this way are split into two equal parts, the first of which will be our In Sample data and the second half will be the Out of Sample data.

2. SEBO then explores all possible variations of the trading strategy, based on the following parameters specified by the user:
   - **Stop loss:** This is the maximum percent loss that can be sustained before the position is liquidated. To limit the number of choices, the user only chooses a maximum integer, so that the tool explores integer values up to the upper boundary.
   - **Holding period:** This is the maximum length of time that stock can be held before it is sold. This is given in terms of trading days per month, with a value that cannot exceed 22. The test tool examines all possible number of days between 1 and the maximum number of holding period specified by the user.
   - **Entry day:** This is the business day that the strategy enters the market in each trading month. All 22 trading days of a month are tried. The user does not control this parameter.
   - **Side:** This is the type of trading strategy, either "long" (profits are to be made when the stock is rising) or "short" (profits are to be made when the stock is falling). Our tool examines both choices of long and short for every combination of other parameters.

3. For each combination of parameters, SEBO computes the Sharpe ratio on the given input time series. When a set of parameters achieves a better Sharpe ratio than the current best set, SEBO records the parameter set and plots the value of the investment. After SEBO has examined all possible combination of parameters, the set of parameters it has on record is the "optimal" variation of the investment strategy.

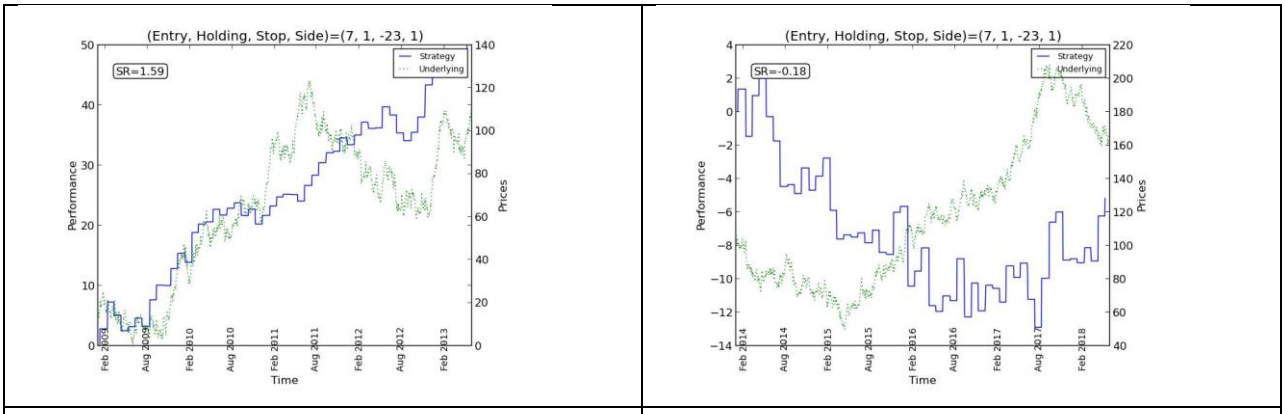
4. The program then generates a second pseudorandom time series, to test the strategy on a different time period (an "Out of Sample" dataset). The "optimal" variation of the investment strategy is applied to this second time series, and a Sharpe ratio is computed.

5. The program then outputs, on the result page, a "movie" showing the progression of the generation of the optimal strategy on In Sample (backtest) data on the left-hand side of the

result page, with the performance of the final, "optimal" strategy on Out of Sample data shown in the graph on the right-hand side of the result page.

4.2. How SEBO is used

The tool has an online form for the user to specify the parameters mentioned above. The values that can be assigned by the user are the maximum holding period, maximum stop loss, length of the backtest, standard deviation of the Gaussian distribution, and seed of the pseudorandom generator. As explained above, the last three parameters control the simulated prices, while the first two parameters control how to exit the market. The SEBO program employs an extremely simple choice for when to enter the market: it simply chooses a fixed trading day of the month to enter the market. It tries all 22 possible choices in this case.



(a) **(b)**

Figure 2: The "optimal" strategy's performance (a) on the In Sample dataset, and (b) on the Out of Sample dataset.

If a user is unsure what parameters to use, she may use the parameters that generated the example in Figure 2, or request the tool to choose a set of random parameters within the acceptable ranges. If one inputs a value for a parameter that is out of the acceptable range, the software uses a preset value for the parameter that falls within the acceptable range. The intent here is to permit the tool to be used by persons with a wide range of expertise in the field, from elementary to advanced.

After the execution of SEBO program, two figures are generated on the output page. In the examples from Figure 2, the green line is the underlying time series, and the blue line shows the performance of the strategy. "SR" denotes the Sharpe ratio. In most runs, the SR of the right-hand graph (i.e., the final strategy on Out of Sample data) is either negative or much lower than the SR of the final left-hand graph (i.e., the "optimal" strategy on In Sample data), indicating that the strategy has been overfit on the In Sample (backtest) data.

In the specific example shown in Figure 2-a, note that the SR of the final optimized strategy, when applied to the input (In Sample) dataset, is 1.59, indicating a fairly promising strategy (the annualized rate of return is 1.59 time the risk undertaken). However, when this same "optimized" strategy is applied to the second (Out of Sample) dataset, as shown in Figure 2-b, the resulting SR is -0.18, indicating a completely ineffective strategy (it is actually somewhat prone to lose money). Even though in both cases, the underlying prices seem to oscillate in the

similar way, on the In Sample data, the investment represented by the blue line goes steadily up, while the same line on the Out Of Sample data on the right goes steadily down. This suggests that the "optimal" strategy's excellent performance on the In Sample dataset was only a statistical fluke – the strategy was optimized to the particular characteristics of that data and had no fundamental "intelligence" to deal with any other dataset.

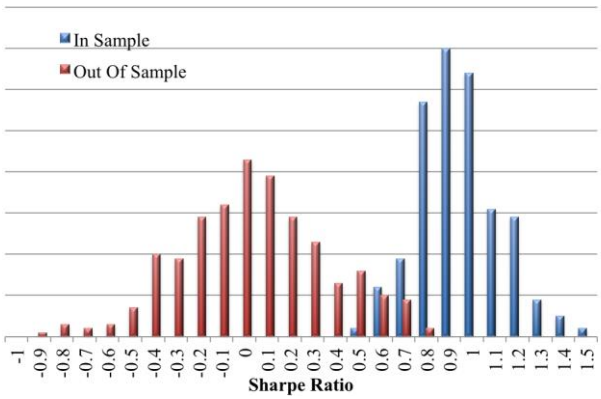


Figure 3: Distribution of Sharpe ratios from 400 test runs with same parameters except the seeds to the random number generator.

4.3. Understanding the results

Since the sample prices are generated with Gaussian distribution centered on zero, we expect the average investment strategy to have a Sharpe ratio of zero. This is indeed the case on the Out of Sample data in 400 test runs with different seeds for the random numbers (see Figure 3). In contrast, on this set of tests, the Sharpe ratios on the In Sample data are centered on 0.9, which is significantly higher than zero.

Half of the test runs use the same set of parameters except the seed for the pseudorandom number generator. These parameter values are: maximum holding period of 25, maximum stop loss percentage of 50, backtest length of 2500 (simulating 10-year worth of daily data), and a standard deviation of 2. Given these parameters, SEBO will investigate 55,000 different parameter combinations on the In Sample data. In other words, only a few tens of thousands of different variants of the proposed investment strategy are examined in each test run. Had more variants been examined, the Sharpe ratio of the tested strategies on In Sample data would doubtless have been even higher. The other half of the test cases use random parameter values for maximum holding period, maximum stop loss percentage, backtest length, and standard deviation. The seeds for random number generator in SEBO vary from 201 to 400.

Keep in mind that we are optimizing only a couple of parameters. Investment strategies often involve many more parameters. If we add another parameter, like a profit taking threshold, the overfit SR will be boosted further, to any desirable level. But our goal is to show that even the simplest of the investment strategies can be easily overfit. An important conclusion is that there is no SR threshold or haircut that can be considered safe.

Although there is much still to be learned about the phenomenon of backtest overfitting, it is clear from running just a modest number of cases that when attempting to produce an "optimal" strategy, it is very hard to avoid backtest overfitting. Indeed, the online tool demonstrates how easily false trading strategies can be derived from purely random data. And, if one does not know how many variations of a strategy have been attempted when developing a strategy, there is no way to know *a priori*, one way or the other, whether the resulting strategy is overfit.

Another conclusion from using the online tool is that the "hold-out" method is not very effective in preventing backtest overfitting. If the web application is run once, it is very likely that the

optimized strategy will perform well on the In Sample dataset but poorly on the Out of Sample dataset. However, if enough cases are tried using the online tool, a strategy that performs well both In Sample and Out of Sample can be discovered. And yet, as before, the strategy cannot have any innate "intelligence," since it is generated based on a pseudorandom dataset.

It should also be emphasized that while this tool was designed to demonstrate the effect of backtest overfitting in mathematical finance, the fundamental underlying principle of statistical overfitting applies very broadly nonetheless. Thus this tool could be easily modified to demonstrate a much broader class of overfitting problems. Indeed, by simply renaming of the input parameters and output results (i.e., renaming the Sharpe ratio, and suitably changing the output plots), one could consider the online tool to be a test of statistical overfitting when attempting to "guess" the future course of any process that can be modeled by a random walk.

5. Conclusion

We have developed an online tool to demonstrate the dangers of backtest overfitting in the mathematical finance field, although, as we emphasized above, the underlying mathematics and software design could easily be considered to be a demonstration of the much broader problem of statistical overfitting of a random walk process.

By using the tool to generate even a modest number of trials, it is immediately clear that it is extremely easy, by using a computer to explore the parameter space of variations of a basic strategy, to "discover" what appears to be an "optimal" trading strategy that gives great-looking performance, based on standard financial performance statistics such as the Sharpe ratio, but yet is completely impotent when presented with any other dataset. The problem, as emphasized above, is that the resulting "optimal" strategy is statistically overfit, since far more variations of the strategy have been tried that can be justified given the size of the input dataset. For this reason, it is actually quite likely that even a modestly sophisticated search process will identify what mistakenly appears to be a promising strategy.

We hope that this research will help investors understand the dangers of backtest overfitting in particular, and selection bias in general. There is still much to be learned about this perplexing phenomenon.

Acknowledgements

The authors thank Bin Dong of the Lawrence Berkeley National Laboratory for his technical assistance in producing this online tool. We also acknowledge helpful suggestions and comments by Jonathan M. Borwein (Priority Research Centre for Computer-Assisted Research Mathematics and Its Applications, University of Newcastle, Australia), Qiji Jim Zhu (Department of Mathematics, Michigan State University) and David Witkin (StatisTrade) for their input. This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program.

References

Ackerman, A. [2014], "SEC finds 'deficiencies' at hedge funds," *Wall Street Journal*, 22 Sep 2014, available at http://online.wsj.com/articles/sec-finds-deficiencies-at-hedge-funds-1411403677.

Bailey, D., J. Borwein, M. Lopez de Prado and J. Zhu (2014): "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-Of-Sample Performance", Notices of the American Mathematical Society, 61(5): 458-471. Available at http://ssrn.com/abstract=2308659

Bailey, D., J. Borwein, M. Lopez de Prado and J. Zhu (2015): "The Probability of Backtest Overfitting", Journal of Computational Finance, forthcoming. Available at http://ssrn.com/abstract=2326253

Bailey, D., M. Lopez de Prado (2012): "The Sharpe Ratio Efficient Frontier", Journal of Risk, 15(2): 3-44. Available at http://ssrn.com/abstract=1821643

Bailey, D., M. Lopez de Prado (2014): "The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting and Non-Normality", Journal of Portfolio Management, 40(5): 94-107. Available at http://ssrn.com/abstract=2460551

Harvey, C. and Y. Liu (2014): "Evaluating Trading Strategies", Journal of Portfolio Management, 40(5): 108-118. http://ssrn.com/abstract=2474755

Harvey, C. and Y. Liu (2015a): "Lucky Factors", working paper. Available at http://ssrn.com/abstract=2528780

Harvey, C. and Y. Liu (2015b): "Backtesting", working paper. Available at http://ssrn.com/abstract=2345489

Harvey, C., Y. Liu and H. Zhu (2015): "… and the Cross-Section of Expected Returns", working paper. Available at http://ssrn.com/abstract=2249314

Beaudan, P. [2013], "Telling the good from the bad and the ugly: How to evaluate backtested investment strategies," working paper, available at http://ssrn.com/abstract=2346600.

Hand, D. (2014): *The Improbability Principle*, Macmillan.

Lopez de Prado, M. (2013): "What to look for in a backtest," working paper. Available at http://ssrn.com/abstract=2308682

Prinz, F., T. Schlange, and K. Asadullah (2011): "Believe it or not: how much can we rely on published data on potential drug targets?", *Nature Reviews Drug Discovery*, 10(9): 712.

Simon, H. et al. (2015): "The Top 500 List". Available at http://top500.org/statistics/perfdevel.