Machine Learning and the Stock Market

Jonathan Brogaard
David Eccles School of Business, University of Utah
Email: Brogaard@Eccles.Utah.edu

Abalfazl Zareei
Stockholm Business School, Stockholm University
Email: Abalfazl.zareei@sbs.su.se

Abstract

Recent advances in machine learning methodologies have improved the usefulness of the technology. This paper examines whether machine learning using only past prices as the input can detect mispricings. Generally searching for mispricings is a slow process and can easily suffer from data-snooping. This paper provides a machine learning algorithm to search for mispricings while controlling for data-snooping. The process generates significant out-of-sample alpha. Overall, the results show that mispricings still exist, but have decreased over time, implying that markets have recently become more efficient.

JEL classification: B26, G12, G14, C58, N20

Keywords: Asset pricing, Anomalies, Mispricings, Machine learning, Big data analysis

Anomalies in asset pricing are cross-sectional or time-series empirical patterns inconsistent with a central asset pricing model. Systematically searching for anomalies is a slow process as there are voluminous amounts of data that can be combined and conditioned. Even if an anomaly is identified it can be due to overfitting. This paper exploits machine learning technology to hasten and regulate the search for anomalies. We do so by utilizing recent developments in machine learning that combines data-mining and optimization techniques while controlling for data-snooping.

We view the search for anomalies from an optimization perspective. The optimization problem is defined by three ingredients: a search space, an objective function, and a set of constraints. In our case the search space is a large set of trading strategies, the objective is to find strategies with high abnormal returns, and the constraint is for the abnormal returns to be statistically significant. The optimization problem has several local optima: there can be several anomalies with high and statistically significant abnormal returns.

To solve the optimization problem we use genetic algorithms. The genetic algorithm machine learning technique adopts the principles of natural evolution and is founded upon the notion of "the strongest survive." It starts with a randomly generated population (a set of strategies) and applies the concept of evolution to generate stronger population sets. It repeats the routine to find the strongest members that survive until the final generation. The innovative feature of the genetic algorithm approach is the way in which the notion of evolution is applied to generate stronger population sets and its ability to find several local optima.

This paper's technique controls for data-snooping. If one searches long enough they can always find an anomaly; however, it may be just a fortunate coincidence. In order to tackle data-snooping issue we take several steps. First, in the optimization procedure we separate the performance of anomalies into two separate periods: a training period and a selection period. The selection period acts as a validation filter to make sure the anomalies found by the algorithm in the training period are not simply a fortunate aberration. Second for those algorithms that perform well in the training and selection periods we evaluate the performance in a third out-of-sample period. We focus on the out-of-sample period. Consistent out-of-sample performance indicates that machine learning can be used to detect asset pricing anomalies.[1] Third, we evaluate the performance of anomalies in other datasets as an additional out-of-sample robustness.

We focus on technical trading rules that generate time-series anomalies. The search space for the optimization procedure is composed of a large set of technical trading rules. The reason behind the search space choice is that it is straightforward to model technical trading rules. The algorithm needs access to a large set of solution candidates and the realm of technical trading rules gives us a computationally inexpensive way of modeling a large set of strategy candidates. The algorithm can be extended or constrained to look for other types of anomalies, such as cross-sectional anomalies or anomalies from fundamental information.

We apply the algorithm to find anomalies in the different NYSE/AMEX volatility decile portfolios between January 1, 1965 to December 31, 2014. We find the algorithm consistently selects trading rules that perform well out-of-sample. The results show that a machine learning approach is able to generate an average four-factor alpha of 25.8% annually for a diversified portfolio of volatility deciles. We also

---

[1] McLean and Pontiff (2016) and Harvey, Liu, and Zhu (2016) have pointed out the concerns of data mining due to the large number of anomalies being discovered in the literature. In response to the data mining concern Harvey et al. (2016) propose a t-statistic of three as a threshold for anomalies' significance. We adopt the Harvey et al approach t-statistic approach.

compare this paper's approach to an alternative static technical trading strategy based on moving averages examined by Han, Yang, and Zhou (2013). Using their set of moving average strategies as an alternative benchmark we find this paper's approach outperforms on average by 21.6% annually. On average the alphas generated by the machine learning algorithm is six times that of the alternative benchmark.

We apply machine learning and big data analysis to financial economics. The algorithm requires a vast amount of computational power. The average time needed to find the optimum trading rules for a diversified portfolio of ten NYSE/AMEX volatility assets for the 40 year sample using a computer with an Intel® Core(TM) CPU i7-2600 and 16 GM RAM is 459.29 days (11,022.97 hours). For one year it takes approximately 11.48 days. We run the analysis on a supercomputer with an Intel Xeon E5-2690v4 CPU and 128 GB memory per node via parallel computing procedures. Compared to 1999 when Allen and Karjalinen (1999) published their algorithm the computing power has increased substantially. Nordhaus (2001) shows that the computing power through time has increased by an average of 55% per year since 1940 with growth post 1980 at around 80% per year. The phenomenal increase in computational power is necessary for the analysis in this paper.

The paper can be viewed as an update to Allen and Karjalainen (1999). We modify their genetic algorithm in five aspects. First, we provide a standardized mechanism to generate technical trading rules. In particular, a trading rule is considered to have four levels where in each level a predetermined set of variables or functions is specified. Since the machine learning algorithm aim is to find local optima, by imposing a structure on the search space we make sure the rules follow a valid structure. Second, the fitness function that determines the appropriateness of trading rules is changed from excess returns to abnormal return. Focusing on abnormal returns ensures we control for return performance driven by systematic risk factors.

Third, the algorithm's search process automatically removes trading rules with statistically insignificant alphas. Fourth, the search process finds a set of optimized rules and not only one rule. Finally, we propose a diversified portfolio of trading rules across several assets. For each asset (volatility decile) we apply the algorithm and find a set of optimized and validated rules. We then diversify equally across all of the rules for all of the assets.

Our analysis is on the weak form of market efficiency. Assuming the Fama and French four factor model is the correct asset pricing model, any alpha we observe can be interpreted as an inefficiency. We compute the average yearly out-of-sample alpha across the ten NYSE/AMEX volatility portfolios for optimized portfolios in a rolling fashion from 1965 and regress the alphas on a time trend variable. We find that the coefficient on the time variable is -0.63, implying that on average each year since 1965 the abnormal return of optimized rules decreases by 0.63%. The decrease in abnormal returns over time is consistent with markets becoming more efficient over time.

This paper makes a number of contributions to the literature. First, we show that machine learning can be used to find mispricings. Second, we do not depend on knowing the anomalies ex-ante. The conventional approach in investigating the performance of anomalies is constrained by the researcher knowing the anomalies, which creates a bias with the ad-hoc specification of anomalies (Chordia, Subrahmanyam, and Tong, 2013; McLean and Pontiff, 2016; Linnainmaa and Roberts, 2018). Using this paper's approach, we do not need to know the anomalies;

we systematically uncover them.

Third, we avoid existing anomaly driven data-snooping. There is a data-snooping concern with already documented anomalies as pointed out by Linnainmaa and Roberts (2018). In the search process, we make sure data-snooping is minimized. Fourth, we can specify the type of market efficiency (weak, semi-strong, or strong) that we want to investigate. For example, since we are searching for time-series anomalies that use information in prices, we are essentially measuring changes in the weak-form of market efficiency. Other uses could be to build a more flexible asset pricing model (Gu, Kelly, and Xiu (2018)) or by adjusting the input information to the algorithm, to measure the semi-strong or strong form of market efficiency.

1. Searching for Anomalies

This section describes the approach to searching for anomalies. We first describe the decision to focus on price path dependent anomalies. Next we describe the machine learning search mechanism in detail. Finally, we discuss the optimization procedure.

1.1. Weak form market efficiency

This paper focuses on finding time-series anomalies based off of past prices, generally referred to as technical trading rules. An example of a technical trading rule is a moving-average strategy (Han et al., 2013; Han, Zhou, and Zhu, 2016). The moving-average strategy functions by producing buy and sell signals based on price patterns. Han et al. (2013) finds that moving average strategies generate positive and statistically significant abnormal returns through time.

We concentrate the search process on technical analysis for three reasons: First, modeling the technical trading rules is straightforward and therefore gives us a computationally inexpensive opportunity to perform a data-mining procedure in discovering inconsistencies. Second, technical analysis encompasses prediction rules with unknown statistical properties using past information. The rules are often developed ad hoc by practitioners. Perhaps the most well-known technical trading rule though has penetrated academia, the momentum strategy. The Jegadeesh and Titman (1993) momentum strategy has become a core asset pricing factor starting with the four-factor Fama and French (Carhart, 1997) model. Third, since trading rules operate by generating buy and sell signals they can be directly applied as a trading strategy. In the next subsections we elaborate on the apparatus of the algorithm.

1.2. Searching Mechanism

Our objective is to find strategies with the highest alphas. The constraints are for the alphas to be positive and statistically significant and for the incurred transaction costs to be lower than a specific level. We formulate the search criteria as an optimization problem:

$$\begin{aligned} & maximize\ alpha(strategy) \\ & \quad subject\ to: \\ & \qquad p-value\ (strategy) < level_{p-value} \\ & \qquad transaction\ cost\ (strategy) < level_{transaction\ cost}, \end{aligned} \tag{1}$$

where $alpha(strategy)$ is the abnormal return of the strategy. We require the strategy to be statistically significant by applying the constraint of p-value being lower than a specified value, $level_{p-value}$. Moreover, $level_{transaction\ cost}$ is the maximum level of transaction costs that we want the strategy to incur. The transaction cost constraint demonstrates the flexibility of the machine learning algorithm.

The optimization process faces several obstacles. First, the objective function is non-differentiable. It is the abnormal return of a technical trading rule; it is a discrete number and depends on the nature of the trading rule under review so we cannot employ the conventional gradient-based methods to find optimal solutions. Second, the search space may have several local optima and we may get trapped in a statistically insignificant anomaly in the search life-span. Third, the size of the search space is large as there are many possible technical trading rules. To overcome these challenges we employ a genetic algorithm approach to operate the anomaly searching mechanism. The genetic algorithm approach is ideal for solving optimization problems with non-differentiable objective functions. Moreover, due to its stochastic nature it is less likely to converge to a local optima, and it is suitable to find optimum solutions in large search spaces. These characteristics make genetic algorithm preferable for the analysis (Allen and Karjalainen, 1999; Neely, Weller, and Dittmar, 1997; Potvin, Soriano, and Vallee, 2004).

*G*enetic algorithms adopt the principles of natural evolution (schema theorem: best observed building block or schema survives) in searching for an optimal solution. It was developed by Holland (1962, 1975) and since then has been applied in various fields such as economics, management science, engineering, and cognitive science. The genetic algorithm generates a sample population of solution candidates. The solution candidates are ranked according to a specific fitness function (objective function). A new population is generated by combining solution candidates according to their relative fitness using crossover and mutation operators. The crossover operator generates new solution candidates by inserting partial characteristics of fitted candidates into new born solution candidates. The mutation operator inserts random changes in the structure of solution candidates to generate new solutions. The algorithm keeps generating new fitted populations until a stop criterion is reached.

Trading Rule Representation

We need to define the set of possible trading rules. Any trading rule representation should have three main characteristics. First, it should cover a large set of viable technical trading rules as we want all available possibilities to be reachable before starting the search process. Second, the trading rules should be feasible. A random integration of functions and operators may result in meaningless rules that only increases the computational expensiveness of the search process. Thus, the trading rules should follow a predefined structure that guarantees the generation of reasonable trading rules. Third, any large set of trading rules that is generated should include well-known technical trading rules.

Our trading rule representation insures these characteristics. We follow Allen and Karjalainen (1999), and adopt a tree-format representation of the solution candidates. However, their representation suffers from the absence of a definitive imposed structure on trading rules resulting in the generation of meaningless trading rules that make the algorithm’s job of finding optimum solutions cumbersome. Improving upon their representation we make sure the generated solution candidates are sensible. Instead of blindly combining functions and operators to construct a trading rule, we make sure the generated trading rules are valid and acceptable (improving the closure property of the solution candidates where all trees are to be synthetically valid composite functions). We ensure validity by imposing a level-based structure on the trading rules where in each level a definitive set of functions and operators are included. A description of trading rules encoding and lists of functions and operators in each level is described in Table 1.

[Table 1 About Here]

The tree-structure has four levels. In the root node, level 1, we use Boolean operators and functions (If-then-else, and, or) that establish the buy or sell signal. In the second level we incorporate relational operators (>, <) that return zero or one values.

In the third level we include real functions (Average, Maximum, Minimum, Median, Lag, Volatility, RSI and Filter). Selecting average, maximum, minimum and lag functions are supported by Neftci (1991), who shows that many trading rules rely on specific patterns of local extremes of past data. The Lag operator captures the change in the price patterns. The volatility function compares the volatility of the underlying variable in the selected days to the volatility across the entire input information length.[2] RSI, the relative strength index, is a technical indicator determining whether a stock is over-bought or over-sold with respect to prices. It is measured by, $RSI = 100 - 100/(1 + RS^*)$, where RS^* is the average number of days that the underlying variable is above its average divided to the average number of days, it is lower than its average. Assuming the variable is the asset's price, an RSI above 70 regards the asset being over-bought and a value lower than 30 suggests the asset is over-sold. The Filter operator is responsible for generating trading signals similar to filter rules in the context of technical analysis. Assuming price is the

[2] The information length is the given input information to the algorithm. As we explain later, we use a total of last 100 days of underlying asset's price and return information.

underlying variable for the operator, the Filter operator takes in two parameters, Pr: a value between -1 and 1 and Days: number of days, and produces $P_{t-Days} + Pr * P_{t-Days}$ where P_{t-Days} is the price value number of Days before today, t. Comparing the price today with the calculated price, the rule generates a trading signal.

Level 4 includes the input parameters to the trading rules. We consider price and return as the input market information that is inserted in the signal generation process.[3] The number of observations inserted into the trading rules depends on the terminal, Days, that specifies the number of days before the current day. Moreover, if the real function Filter is chosen in level 3 we also include Pr, a random number between -1 and 1 as an input to the trading rule.

We use the decoding procedure to generate distinct buy and sell signals. Figure 1 presents an example of a possible trading rule.

[Figure 1 About Here]

The Figure 1 example trading rule consists of two branches. The left one generates a short-selling signal. According to the rule we short-sell the asset if the price is lower than the average of prices during the last 80 days. The right branch is responsible for the buy signal. We buy the underlying asset if the price today is higher than the average prices in the last 20 days. In the case when both branches generate the same signal (buy or sell) we hold the risk-free asset (i.e. out-of-market). The example in Figure 1 presents a case where in the root node we employ the "If-Then-Else" operator. A more complicated case of trading rules can be obtained by employing "and" and "or" operators in the root node.

[3] Trading volume is an additional variable that can provide useful information into the trading rule mechanism (Blume, Easley, and O'hara, 1994; Grundy and McNichols, 1989). For simplicity in this paper we focus exclusively on information in prices.

The mechanism is flexible in generating technical trading rules. The four-level representation of trading rules can generate 130 million possible structures using data only from the last 100 days. In addition, the mechanism is able to produce well-known technical trading rules, such as filter, moving average, support and resistance, and breakout rules.

Fitness Value

Having established a set of possible trading rules, we next need to calculate the fitness value. The fitness values drive the search direction. Looking at the search for anomalies as an optimization problem, the fitness values are computed by combining objective functions and the constraints. We quantify the fitness value as follows: Considering a trading rule, a zero-cost portfolio is computed by subtracting the trading rule's return ($\widetilde{R}_t$) from the return of the buy-and-hold strategy (R_t). The portfolio's return is $PR_t = \widetilde{R}_t - R_t$. Next, we regress the zero-cost portfolio return on the Fama and French four-factor returns: [4]

$$PR_t = \alpha + \beta_{MKT} r_{MKT} + \beta_{SMB} r_{SMB} + \beta_{HML} r_{HML} + \beta_{MOM} r_{MOM} + \epsilon_t, \quad (2)$$

where r_{MKT}, r_{SMB}, r_{HML}, and r_{MOM} are the returns on market, size, value and momentum portfolios. We apply Newey and West (1987) standard errors. α measures the abnormal risk-adjusted return and is the variable that the algorithm aims to maximize, conditional on it being statistically significant. From Equation (2) we obtain the value of α and its corresponding *p*-value.

While the primary fitness value is the alpha value we also want the alpha value to

[4] Oher asset pricing models such as CAPM, three-factor or five-factor model can also be considered for the objective function. We consider alternative models in the robustness tests.

be statistically significant. Consequently, we adopt the algorithm to manipulate the fitness values. In order to consider the *p*-values in evaluating the relative fitness of candidate solutions, we force the fitness values to be equal to $-M$ (M being a large value) when *p*-values are higher than 0.01. The trading rules with non-statistically significant alphas or high transaction costs are eliminated from the candidate solutions in the genetic optimization process.

We adopt a similar procedure to include transaction cost in the optimization procedure. Following Balduzzi and Lynch (1999), Lynch and Balduzzi (2000), Han (2006) and Han et al. (2013), transaction costs are assumed to be incurred when trading the underlying asset (long or short). When trading the 30-day T-bill we assume no costs. We calculate the breakeven transaction cost (BETC) value that makes the average return of the trading strategy's return (R_t) equal to zero. Balduzzi and Lynch (1999) use one basis point (bp) and 50 basis points (bps) as the lower and upper bounds for transaction costs. In order to show the algorithm's ability in finding profitable tradeable anomalies we require the search to find trading rules with a BETC higher than 25 bps.[5]

Transaction costs and the p-value constraint are included in the searching mechanism. The fitness value for trading rule i is computed as:

$$fitness_i = \begin{cases} \alpha_i, & if\ p-value_i \leq 0.01\ and\ BETC_i \geq 25\ bp \\ -M, & Otherwise \end{cases},$$

(3)

where α_i is the alpha value computed for trading rule i and M is an arbitrarily large

[5] Lynch and Balduzzi (2000) consider the same level of transaction cost. In addition, Sadka and Scherbina (2007) investigate the TAQ dataset and estimate 25 bps as the average effective spread for a typical stock and a typical trade in the TAQ dataset.

number. The form of the fitness value realization in the genetic algorithm procedure ensures survival of trading rules with abnormal returns that are statistically significant at least at 1% level (that equates to a t-statistic higher than 2.6 in this paper's setting) while their BETC is higher than 25 bps.

Optimization Procedure

Having designed the search mechanism, the next step is to optimize the testing procedure. We start by creating a set of 500 randomly generated trading rules from the universe of possible trading rules, called the primary set. The primary set represents the raw trading rules before any optimization procedure is taken place. The population evolves via genetic algorithm by producing new populations while ensuring the fittest survives.

To create the new evolved populations of trading rules we employ crossover and mutation operators. The crossover operator aims to produce new trading rules by using the characteristics of the existing population members. In crossover we randomly select two rules ("Parents" in the genetic algorithm context) in the existing population and switch one branch from the buy-side of one rule with a branch from a sell-side of another trading rule. The branches are chosen randomly in both rules. Two new rules are generated. In the genetic algorithm literature the new rules are referred to as Offsprings. Appendix A provides an example of crossover and mutation operators.

The mutation operator aims to preserve the diversity in the population. We apply two different mutation operation techniques. First, we follow Allen and Karjalainen (1999) and implement mutations by doing a crossover between a randomly chosen rule in the existing population and a new generated trading rule. Using the newly generated rule introduces additional diversity in the population. Second, we apply another form of mutation operation by simply inserting into the population newly generated rules. The two mutation operations increase genetic diversity in the procedure.

We apply the crossover and mutation operations to the existing population in order to generate new sets of trading rules. We generate a population using the crossover operator. Each crossover operation generates two new rules. We apply the crossover operator until we generate the same number of rules as in the population set. In addition, we use the mutation operator to generate another new population. We generate two new populations where the new populations have both the characteristics of the existing (survived) population and also random characteristics with the objective of preserving the genetic diversity. The new generated populations are merged with the existing one and the fitness values are evaluated. The trading rules are sorted based on the highest fitness values and the new population consisting of the 500 best members is selected.

Applying the algorithm to a specific time-period potentially introduces a data-snooping problem: any anomalies that we discover may only be an anomaly for those data points used in the optimization procedure. To address the data-snooping issue we divide the time span of the algorithm's information input in to two periods: a training period and a selection period. The selection period acts as a validation period to ensure the anomalies found by the genetic algorithm in the training period are not simply from data-snooping. We start by searching along the training period to construct new populations; after selecting the fittest trading rules in the training period we re-evaluate them in the selection period. The successful rules from the selection period are added to a final dataset, the Final Set. In the next generations, the members in the Final Set are updated until the stopping criterion is reached. Table 2 presents the genetic algorithm procedure in detail.

[Table 2 About Here]

Two of the main factors in determining the performance of the algorithm are population size, |POP|, (number of rules in the population set), and number of generations, |GEN|, (the number of times that the population is reproducing using cross-over and mutation operators). A larger population size results in a higher number of optimized trading rules after the completion of optimization procedure. However, a larger population size makes the optimization more computationally expensive. The choice of population size is a trade-off between a greater number of optimized solution candidates and computational intensity. We follow Allen and Karjalainen (1999) and choose the population size to be 500, a number large enough to result in a meaningful evaluation of the algorithm performance while being computationally manageable. In robustness tests we run the experiments with other population sizes (100, 1000, 2000) and the results are economically similar.

The choice of the number of generations, |GEN|, also carries a trade-off between computational expensiveness of the procedure and optimized characteristics of the solution candidates. From the Darwinism perspective more generations means more competition, survival, and reproduction in the evolution process. In the optimization procedure a higher |GEN| results in finding better-performing rules, but it also means more computational expense. We report the results using 20 generations. In robustness tests we use 50 and 100 generations and the results are qualitatively similar.

2. Can Machine Learning Find Mispricings?

In this section we examine the algorithm's performance. We begin by describing the data and a number of implementation choices. Next, we report the results for the analysis on ten separate assets. We then allow the algorithm to create a portfolio across the assets.

2.1. Data and Implementation

We use ten NYSE/AMEX volatility decile portfolios as the test assets and evaluate performance between July 1, 1965 to December 31, 2014[6]. We consider a five year training period and a five year selection period.[7] The rolling analysis procedure is as follows: we start from 1965, use the five-year training period (1965-1969) and five-year selection period (1970-1974) as the input to the algorithm. After optimization, we test the performance of strategies in the one year out of sample (1975). Next, we roll the input data by one year and use the five-year training period (1966-1970) and five-year selection period (1971-1975) as input to the algorithm and evaluate the performance of the algorithm's output rule in the one year ahead out of sample period (1976). We repeat the procedure until 2014. In addition, in each rolling window we perform the analysis 20 times to make sure the results are not driven by an auspicious one time run of the algorithm.

Our main interest is in the performance of the Final set, which contains the set of optimized technical trading anomalies. For the baseline benchmark we use the Fama and French four factor model. As a horse race for a weak-form efficient market

[6] Later as a robustness check we consider NYSE/AMEX/NASDAQ size decile portfolios.
[7] In untabulated results we also consider 3, 8, and 10 year training and selection periods and the results are qualitatively similar.

benchmark alternative we use the performance of the Han et al. (2013) moving average strategy. Han et al. (2013) show that moving average (MA) strategies act as anomalies with respect to CAPM, three and four factor asset pricing models and moreover, they realize low transaction costs.[8] They consider a list of specific moving average strategies with lag lengths from 3 to 200. We follow the same procedure and generate a simulated version of moving average strategies with lags between 3 to 100 and we call it the Moving-Average Set.[9]

In addition, we consider two other horse races. The first is the performance of the Primary set. Recall that the Primary set is the initial randomly drawn strategies over which we optimize to produce the Final set. If the optimization has no value then the Final set and the Primary should perform similarly.

2.2. Asset-by-Asset Performance

Table 3 reports the average annualized Out-of-Sample four-factor alpha values for the four sets of trading rules across volatility decile portfolios.

[Table 3 About Here]

The column *N* is the number of unique technical trading rules. For the Final Set *N* is the average number of unique trading rules found in each rolling period and each simulation. The other columns report statistics on the distribution of the performance of the trading rules. Table 3 has four key insights.

---

[8] Han et al. (2013) use 10 NYSE/AMEX volatility decile portfolios between July 1, 1963 to December 31, 2009 to show the well-performance of moving average (MA) strategies. In an untabulated analysis, we compare the optimized rules with MA strategies in the same exact dataset and the same periods as in Han et al. (2013). The results are economically similar.

[9] We choose 100 because in the baseline optimization we use the information up to 100 days before. The results are economically similar when considering 200 days.

First, the Final Set generates higher average alphas across volatility decile portfolios than the other trading rule sets. For example, in the lowest volatility decile portfolio the Final Set generates a 14.27% annualized abnormal return while the Moving-Average Set and Primary Set generate abnormal returns of 4.76% and -1.26%, respectively. Second, as the asset volatility portfolio increases average alpha values increase for the rules in the Final Set and the Moving-Average Set. For example, the Final Set in the lowest volatility decile portfolio generates 14.27% and for the highest volatility decile portfolio it generates 22.98% (deciles 4 – 9 generate higher alphas).

Third, comparing the average alphas generated by the Primary Set and the Final Set shows that the algorithm adds value. The Final Set contains strategies that generate positive abnormal returns while the Primary Set does not. For instance, in the lowest volatility decile portfolio, the Primary Set generates an average of -1.26% and the Final Set generates an average of 14.27%, showing that the algorithm successfully optimizes. Fourth, the number of trading rules in the Final Set shows that the algorithm finds fewer number of rules than the possible population set (500 rules). For the lowest volatility portfolio the algorithm finds only 332 unique trading rules that satisfy the search criteria.

2.3. Portfolio Performance

Instead of investing in each asset separately we can more realistically create a portfolio. Here we consider a case where we diversify across all of the volatility decile portfolios. The analysis assumes an investor uses the algorithm, finds optimized rules specific for each asset, and invests equally across the rules. The summary statistics for the out-of-sample results are presented in Table 4.

[Table 4 About Here]

A diversified portfolio across the rules results in an average annualized alpha of 25.8%, which is higher than the Moving Average Set's alpha of 4.2% and the Primary Set's alpha of -1.3%.

While the overall average statistics are useful to determine the overall performance of the different Sets, the time series dynamics is also of interest. Figure 2 presents the average annualized out-of-sample alphas for the different sets of trading rules in a diversified portfolio across volatility decile assets through time.

[Figure 2 About Here]

The alpha values for the Final Set through time shows that the algorithm produces positive alphas most years in the sample. In addition, the Moving-Average Set usually generates positive returns through time, albeit at a lower level than the Final Set, and with more frequent and larger negative alpha years. The Primary Set consistently generates near zero or negative abnormal returns through time. Not surprisingly, blindly investing using randomly generated rules does not outperform the Fama and French three-factor benchmark.

The Final Set in Figure 2 appears to have a downward slope to its alpha generation over time. To test whether there is a statistically significant negative time trend we perform a simple OLS regression. We construct a time trend variable, $year_t$, which starts from one in 1975, the first year of out-of-sample results, and increases by one unit until 2015. We compute the average out-of-sample alpha values,

$AvgAlpha^t$ in the Final Set for each out-of-sample year. We regress the average yearly alpha values on the time trend variable. The results are presented in Table 5.

[Table 5 About Here]

The first row is the regression using all the volatility assets. The next ten rows test each of the volatility assets separately. We focus on the overall result. The estimated coefficient on the time trend when the dependent variable is the average alphas in the Final Set is negative and statistically significant. The negative coefficient of -0.63 value implies that as each year goes by the performance of a set of optimized anomalies decreases yearly by 0.63%. As the techniques we employ have been available and likely utilized since the early 2000s it is feasible that the decreasing time trend is due to a one-time shift, not a gradual slope. It is beyond the scope of this paper to try and disentangle the functional form of the time trend. The take-away we make is that even in recent times the adoption of machine learning can find mispricings.

Why is there a negative trend in the average alpha values for the optimized set of anomalies through time? The algorithm's job is to find the best set of anomalies in each period and we show that the abnormal return that can be achieved in a particular time period with the trading rule structure (functions and input variables) that is imposed has declined. The decline suggests that markets have become more efficient over time. That is, fewer mispricings exist that violate weak-form of market efficiency.

Next, we examine the risk-return trade-off by calculating the Sharpe ratio on the raw returns (not alphas). We compute the average value of the out-of-sample Sharpe ratios of rules across the simulations for a diversified portfolio each year. Figure 3 reports the results.

[Figure 3 About Here]

The graph shows that using the machine learning algorithm can lead to sizeable positive Sharpe ratios through time that is consistently higher than one and in the most recent three years has been near 1.5. Not only does the algorithm produces a sizeable alpha, it has an attractive risk-return tradeoff.

Next we test whether the trading strategy is implementable or whether transaction costs nullify the alpha. We consider a one-sided transaction cost of 5 basis points. For each transaction that occurs we subtract 5 basis points from the return. As we move between a long position in the asset, holding the risk-free asset, and a short position in the asset we subtract a reasonable transaction cost. Figure 4 presents the results for abnormal returns after transaction costs for a diversified portfolio across volatility decile portfolios.

[Figure 4 About Here]

The figure shows that even after accounting for transaction costs, the profitability of the optimized trading rules still remains.

3. Characterizing the Trading Signals

This section evaluates the types of trading rules the search algorithm selects. First, we look at the direction of the positions the portfolios hold over time. Next, we evaluate whether the buy or sell side signals drive performance. We also look into the different types of signal functions generated. Before examining the portfolio allocation, we characterize how diverse are the selected trading strategies. We conclude by observing how trading rules evolve across the volatility deciles over time.

3.1 Signal Direction

We begin by analyzing the signal generation outcomes by evaluating the signal direction. We do not require the algorithm to be long or short and so the search algorithm has flexibility in its investment direction. We track the frequency of buy, sell and hold risk-free signals generated by the optimized rules. As an example of what we count as a buy/sell signal, suppose a rule generates signals as follows in five consecutive days: [buy-buy-buy-sell-sell]. Following the signals, we buy the underlying asset in day 1, hold it for three days and then in day 4 we change the position to a short-sell one. In the example we count one buy signal and one sell signal. We only count the signals that require an action from the investor. For each rule in the out-of-sample period we count the number of buy, sell, and out-of-market signals across all of the simulations. The results are presented in Figure 5.

[Figure 5 About Here]

The figure produces a few key takeaways. First, the frequency of buy, sell, and neutral signals are relatively stable over the sample period. Second, the buy signals make up roughly 40% of the signals and the sell and hold risk-free signals each make up about 30% of the actionable signals. Next, we examine whether the performance of the trading rules comes from both the long and short positions or if one dominates the other.

In Table 6 we look at the return summary statistics across the volatility portfolios generated from the buy, sell and hold-risk free signals.

[Table 6 About Here]

The values are the $(t+1)^{th}$ day returns conditioned on the type of signals (buy, hold, sell) at time t averaged over the out-of-sample years and the simulations. For deciles 1 – 8 long signals make up over 50% and short signals make up less than 25%. In decile 9 and 10, the long signal is less than 50% and the short signal occurs more than 25% of the time.

Optimized rules for the lowest and highest volatility decile portfolio generates on average a mean return of 0.113% daily (28.48% annually) and 0.245% (61.74% annually), respectively. The corresponding value for a simple buy and hold strategy is 0.049% and 0.183% average daily return for the lowest and highest volatility decile portfolios, respectively.

The average next day return across all the strategies for the buy signals is 0.168% and 0.245% daily for the lowest and highest volatility decile portfolios, respectively. The average next day return conditioned on the short-selling signals is 0.130% and 0.197% daily for the lowest and highest volatility decile portfolios, respectively. Overall, the results show that both the short and long signals predict future returns and contribute to the search mechanism's success.

3.2 Signal Functions

We further explore the types of functions and variables employed in the trading rules. Do the optimized rules tend to capitalize on a particular function or variable in their structure? Table 7 summarizes the characteristics of the trading rules in the

Final Set when the underlying asset is the lowest and highest NYSE/AMEX volatility decile portfolios.[10] All the values are averaged over the rolling out-of-sample windows over the 20 simulations.

[Table 7 About Here]

We focus on the results for the lowest volatility decile portfolios (the highest volatility decile portfolios follow a similar pattern). The maximum number of times a function or real variable (Price or Return) can be used is 1884 (when each side of trading rules is a two-branch tree); However, it is possible for the rules to have only one branch in the buy and sell side. The functions Minimum and Median are used the most. Price is used more than Return as an input to the optimized trading rules. The results also suggest that there is no single function or variable that dominates. We explore the diversity of trading rules in the next subsection.

3.3 Strategy Diversity

The previous subsection suggests that the trading rules depend on a variety of signals. Here we directly test the diversity of the trading rules. Specifically, we examine whether the strategies in the Final Set generate the same set of signals. Since similar signals result in similar returns across the strategies, we focus the analysis on the returns. Table 8 reports the summary statistics for the pair-wise correlation values of the out-of-sample returns for the optimized trading rules (the Final Set).

[10] Because of the large number of optimized rules generated we are unable to list them.

[Table 8 About Here]

A correlation close to one implies the same signal generation procedure along the optimized trading rules and that the algorithm uses the same form of signal generation procedure. Lower correlation values suggests the optimization procedure finds anomalies in various forms. On average the correlation among strategies' returns is approximately 0.65. The moderate average pairwise correlation suggests the usefulness of the optimization process in finding a diversified portfolio of anomalies.

Figure 6 presents the boxplot for correlation values through time.

[Figure 6 About Here]

The correlation values across trading rules are computed for each out-of-sample year from 1975 and pulled together across simulations. We draw the boxplot for each out-of-sample year. Figure 6 shows that the algorithm finds a variety of trading rules throughout the sample period.

3.4 Portfolio Allocation

Before testing the robustness of the machine learning algorithm, we consider how the algorithm allocates resources across the ten volatility deciles. Recall that the portfolio allocation is equally distributed based on the number of trading rules that survive in the Final Set for each volatility decile. Thus, asking how the algorithm allocates resources across the volatility deciles is the same as asking how many trading rules in each volatility decile make it into the Final Set. Figure 7 presents the weight allocation across volatility decile portfolios through time.

[Figure 7 About Here]

Figure 7 shows that through most of the sample we allocate weights equally across the volatility deciles. Each asset gets about 10% each year. However, starting in 2005 the weight allocation across decile portfolios changes. After 2005 the lowest and highest volatility decile portfolios are the main investment vehicles. For example, in 2014, 35% of trading rules are from the lowest volatility decile portfolio and another 30% are from the highest volatility decile. While explaining the stark shift in portfolio allocation after 2005 is beyond the scope of this paper, we find it to be a fascinating observation.

4. Robustness

This section addresses three possible concerns. First, we conduct a placebo test to show the results are not driven by some mechanical regression process or a fortunate adaptation of rules to the data. Second, we repeat the analysis on a different datset to ensure that the results do not only hold in a specific setting. Third, we consider other asset pricing models to test whether the anomalies we find can be explained by other factor models.

4. 1 Placebo test

It may be that the results are driven by some mechanical regression process or simply a fortunate adaptation of rules to the data. To address this concern we run the following experiment: we scramble the dataset along the time dimension and apply the optimization procedure to the scrambled data. Any predictability from prices should be broken in the dataset and so applying the optimization procedure should be unsuccessful in identifying trading rule anomalies. We use the ten NYSE/AMEX volatility decile portfolios from July 1, 1965 to December 31, 2015. We assign a random number to each time period and sort the data by the random number. The optimization procedure is applied to the scrambled data using five year training and five year selection periods. We consider the population size of 500 and the generation number of 20. The results are presented in Table 9.

[Table 9 About Here]

The average out-of-sample alphas are negative consistent with expectations. If the performance of the optimization procedure is a result of detecting trends, as we argue, scrambling the data causes the optimization procedure to be unsuccessful. As expected, we find that the Final Set consistently produces negative alphas.

4.2 Choice of underlying asset

Another possibility is that the search procedure performs well because of the choice of underlying asset. To test the role of the underlying asset we repeat the analysis but on the NYSE/NASDAQ/AMEX size decile portfolios.

We first report the Kendall rank correlation coefficients between size decile and volatility decile portfolio to ensure the size and volatility portfolios are sufficiently unrelated. Table 10 Panel A reports the correlation values.

[Table 10 About Here]

The correlations between the portfolios are scattered between 0.31 and 0.78. Overall, the rank correlation between the size decile and volatility decile portfolios are low to moderate so using the size deciles gives us a new setting for testing the algorithm's ability to recognize patterns.

We repeat the machine learning analysis on the size decile portfolios as the alternative dataset. We use the time period January 1, 1965 to December 31, 2014 and use the same search procedure (five years of training and five years of selection period). The results are presented in Table 10: Panel B for the out-of-sample period from January 1, 1975 to December 31, 2014.

Table 10 Panel B presents the summary statistics for annualized alpha values for the Primary Set, Final Set, and Moving Average Set. The Primary Set again performs poorly consistently generating negative alpha values across the size decile portfolios. The moving average strategy performance improves as we move towards smaller decile portfolios, consistent with the results of Han et al. (2013). The Final Set performs better than the moving average strategies across the size portfolios.

The law rows of Table 10 Panel B show the results for a diversified portfolio of size decile portfolios. The average alpha for the Final Set is higher than the benchmark set of Moving Average Set and the Primary Set. When diversifying across the size decile portfolios the moving average strategy result in an average annualized alpha of 9.97% while the set of improved rules generate an alpha of 32.32%.

4.3 Choice of Asset Pricing Model

We use the Fama and French four-factor asset pricing model as the benchmark to calculate alphas. It is possible the four-factor model is the wrong model and the results are driven by some unmodeled risk factor. We re-evaluate the out-of-sample performance using CAPM, three-factor and five-factor asset pricing models. The Fama and French three-factor model consists of market (MKT), size (SMB) and value (HML) portfolios (Fama and French, 1993). Fama and French (2015) propose a five-factor asset pricing model with market (MKT), size (SMB), value (HML), profitability (RMW), and investment (CMA) factors. We obtain the data on factors from Kenneth French's website. We repeat the analysis in Table 4 with the alternative asset pricing models. The results are reported in Table 11.

[Table 11 About Here]

The optimized rules continue to consistently generate abnormal returns and to perform better than the other sets of rules.

5 Conclusion

Uncovering anomalies is an important research topic in asset pricing. We argue that using machine learning is useful as it can accelerate the search for anomalies. This paper provides an algorithm based on machine learning to search for anomalies while controlling for data-snooping. The baseline algorithm is designed to look for anomalies with positive and significant four-factor alphas. Applying the algorithm to the ten NYSE/AMEX volatility decile portfolios, we show that it successful finds time-series anomalies that perform better than the benchmark of a set of moving-average strategies.

The algorithm is successful in finding anomalies that are persistent out-of-sample. We use the algorithm to quantify changes in market efficiency through time. In recent years the profitability of weak-form market efficiency anomalies is lower than past decades.

More generally, this paper is on the application of big data exploration in financial economics. Big data analysis comes with the potential for data-snooping, p-hacking or data-dredging (see Harvey, 2017). The goal in this paper is to show that big data paired with sophisticated computational techniques can be informative in financial economics.

References

Allen, Franklin, and Risto Karjalainen, 1999, Using genetic algorithms to find technical trading rules, *Journal of Financial Economics* 51, 245-271.

Bajgrowicz, Pierre, and Olivier Scaillet, 2012, Technical trading revisited: False discoveries, persistence tests, and transaction costs, *Journal of Financial Economics* 106, 473-491.

Baker, Malcolm, and Jeffrey Wurgler, 2007, Investor sentiment in the stock market, *Journal of Economic Perspectives* 21, 129-151.

Balduzzi, Pierluigi, and Anthony W Lynch, 1999, Transaction costs and predictability: Some utility cost calculations, *Journal of Financial Economics* 52, 47-78.

Banz, Rolf W, 1981, The relationship between return and market value of common stocks, *Journal of Financial Economics* 9, 3-18.

Barberis, Nicholas, Andrei Shleifer, and Robert Vishny, 1998, A model of investor sentiment, *Journal of Financial Economics* 49, 307-343.

Blume, Lawrence, David Easley, and Maureen O'hara, 1994, Market statistics and technical analysis: The role of volume, *Journal of Finance* 49, 153-181.

Brown, David P, and Robert H Jennings, 1989, On technical analysis, *Review of Financial Studies* 2, 527-551.

Carhart, Mark M, 1997, On persistence in mutual fund performance, *Journal of Finance* 52, 57-82.

Cespa, Giovanni, and Xavier Vives, 2012, Dynamic trading and asset prices: Keynes vs. Hayek, *Review of Economic Studies* 79, 539-580.

Chan, Louis KC, Yasushi Hamao, and Josef Lakonishok, 1991, Fundamentals and stock returns in japan, *Journal of Finance* 46, 1739-1764.

Chen, Nai-Fu, Richard Roll, and Stephen A Ross, 1986, Economic forces and the stock market, *Journal of Business* 59, 383-403.

Chinco, Alexander, Adam D Clark-Joseph, and Mao Ye, 2017, Sparse signals in the cross-section of returns, J*ournal of Finance*, forthcoming.

Chordia, Tarun, Avanidhar Subrahmanyam, and Qing Tong, 2014, Have capital market anomalies attenuated in the recent era of high liquidity and trading activity?, *Journal of Accounting and Economics* 58, 41-58.

Cochrane, John H, 1999, Portfolio advice for a multifactor world, *Economic Perspectives Federal Reserve Bank of Chicago* 33, 59-78.

Cohen, Randolph B, Paul A Gompers, and Tuomo Vuolteenaho, 2002, Who underreacts to cash-flow news? Evidence from trading between individuals and institutions, *Journal of Financial Economics* 66, 409-462.

Da, Zhi, 2009, Cash flow, consumption risk, and the cross-section of stock returns, *Journal of Finance* 64, 923-956.

Daniel, Kent, David Hirshleifer, and Avanidhar Subrahmanyam, 1998, Investor psychology and security market under-and overreactions, *Journal of Finance* 53, 1839-1885.

Daniel, Kent, and Sheridan Titman, 2006, Market reactions to tangible and intangible information, *Journal of Finance* 61, 1605-1643.

De Long, J Bradford, Andrei Shleifer, Lawrence H Summers, and Robert J Waldmann, 1990, Noise trader risk in financial markets, *Journal of Political Economy* 98, 703-738.

Fairfield, Patricia M, J Scott Whisenant, and Teri Lombardi Yohn, 2003, Accrued earnings and growth: Implications for future profitability and market mispricing, *The Accounting Review* 78, 353-371.

Fama, Eugene F, and Kenneth R French, 1992, The cross-section of expected stock returns, *Journal of Finance* 47, 427-465.

Fama, Eugene F, and Kenneth R French, 1993, Common risk factors in the returns on stocks and bonds, *Journal of Financial Economics* 33, 3-56.

Fama, Eugene F, and Kenneth R French, 1996, Multifactor explanations of asset pricing anomalies, *Journal of Finance* 51, 55-84.

Fama, Eugene F, and Kenneth R French, 2015, A five-factor asset pricing model, *Journal of Financial Economics* 116, 1-22.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2017, Taming the factor zoo. Working Paper.

French, Kenneth R, G William Schwert, and Robert F Stambaugh, 1987, Expected stock returns and volatility, *Journal of Financial Economics* 19, 3-29.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani, 2001, The elements of statistical learning, volume 1 (Springer series in statistics New York).

Grundy, Bruce D, and Maureen McNichols, 1989, Trade and the revelation of information through prices and direct disclosure, *Review of Financial Studies* 2, 495-526.

Gu, Shihao, Bryan T. Kelly, and Dacheng Xiu, 2018, Empirical asset pricing via machine learning, Working Paper.

Han, Yufeng, 2006, Asset allocation with a high dimensional latent factor stochastic volatility model, *Review of Financial Studies* 19, 237-271.

Han, Yufeng, Ke Yang, and Guofu Zhou, 2013, A new anomaly: The cross-sectional profitability of technical analysis, *Journal of Financial and Quantitative Analysis* 48, 1433- 1461.

Han, Yufeng, Guofu Zhou, and Yingzi Zhu, 2016, A trend factor: Any economic gains from using information over investment horizons?, *Journal of Financial Economics* 122, 352- 375.

Harvey, Campbell R, 2017, Presidential address: The scientific outlook in financial economics, *Journal of Finance* 72, 1399-1440.

Harvey, Campbell R, Yan Liu, and Heqing Zhu, 2016, and the cross-section of expected returns, *Review of Financial Studies* 29, 5-68.

Haugen, Robert A, and Nardin L Baker, 1996, Commonality in the determinants of expected stock returns, *Journal of Financial Economics* 41, 401-439.

Holland, John H, 1962, Outline for a logical theory of adaptive systems, *Journal of the ACM* 9, 297-314.

Holland, John H, 1975, Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press.

Hong, Harrison, and Jeremy C Stein, 1999, A unified theory of underreaction, momentum trading, and overreaction in asset markets, *Journal of Finance* 54, 2143-2184.

Jegadeesh, Narasimhan, and Sheridan Titman, 1993, Returns to buying winners and selling losers: Implications for stock market efficiency, *Journal of Finance* 48, 65-91.

Karapandza, Rasa, and Jose M Marin, 2014, The rate of market efficiency, Working Paper.

Kunh, Thomas, 1962, The structure of scientific revolutions. Chicago: Univ. Press, Chicago.

Linnainmaa, Juhani T, and Michael R Roberts, 2018, The history of the cross section of stock returns, *Review of Financial Studies* 31, 2606-2649.

Linter, John, 1965, The valuation of risky assets and the selection of risky investments in stock portfolios and budget constraints, *Review of Economics and Statistics* 47, 13-37.

Lynch, Anthony W, and Pierluigi Balduzzi, 2000, Predictability and transaction costs: The impact on rebalancing rules and behavior, *Journal of Finance* 55, 2285-2309.

McLean, R David, and Jeffrey Pontiff, 2016, Does academic research destroy stock return predictability?, *Journal of Finance* 71, 5-32.

Merton, Robert C, 1980, On estimating the expected return on the market: An exploratory investigation, *Journal of Financial Economics* 8, 323-361.

Muth, John F, 1961, Rational expectations and the theory of price movements, *Econometrica* 29, 315-335.

Neely, Christopher, Paul Weller, and Rob Dittmar, 1997, Is technical analysis in the foreign exchange market profitable? A genetic programming approach, *Journal of Financial and Quantitative Analysis* 32, 405-426.

Neely, Christopher J, David E Rapach, Jun Tu, and Guofu Zhou, 2014, Forecasting the equity risk premium: the role of technical indicators, *Management Science* 60, 1772-1791.

Neftci, Salih N, 1991, Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of "technical analysis", *Journal of Business* 64, 549-571.

Newey, Whitney K, and Kenneth D West, 1987, Hypothesis testing with efficient method of moments estimation, *International Economic Review* 28, 777-787.

Nordhaus, W., 2001, The progress of computing, Working Paper.

Pontiff, Jeffrey, 1997, Excess volatility and closed-end funds, *American Economic Review* 87, 155-169.

Pontiff, Jeffrey, 2006, Costly arbitrage and the myth of idiosyncratic risk, *Journal of Accounting and Economics* 42, 35-52.

Pontiff, Jeffrey, and Artemiza Woodgate, 2008, Share issuance and cross-sectional returns, *Journal of Finance* 63, 921-945.

Potvin, Jean-Yves, Patrick Soriano, and Maxime Vallee, 2004, Generating trading rules on the stock markets with genetic programming, *Computers & Operations*

Research 31, 1033-1047.

Rosenberg, Barr, Kenneth Reid, and Ronald Lanstein, 1985, Persuasive evidence of market inefficiency, *Journal of Portfolio Management* 11, 9-16.

Sadka, Ronnie, and Anna Scherbina, 2007, Analyst disagreement, mispricing, and liquidity, *Journal of Finance* 62, 2367-2403.

Schwert, G William, 2003, Anomalies and market efficiency, *Handbook of the Economics of Finance* 1, 939-974.

Sharpe, William F, 1964, Capital asset prices: A theory of market equilibrium under conditions of risk, *Journal of Finance* 19, 425-442.

Titman, Sheridan, Kuo-Chiang Wei, and Feixue Xie, 2004, Capital investments and stock returns, *Journal of Financial and Quantitative Analysis* 39, 677-700.

Treynor, Jack L, and Robert Ferguson, 1985, In defense of technical analysis, *Journal of Finance* 40, 757-773.

Figure 1. Trading rule example

This figure shows an example of a trading rule. Following the rule means we buy the asset when the price today is higher than the average of prices in the last 20 days and short-sell the asset when the today's price is lower than the average of prices in the last 80 days. We hold the risk-free asset when both or none of the conditions are satisfied.

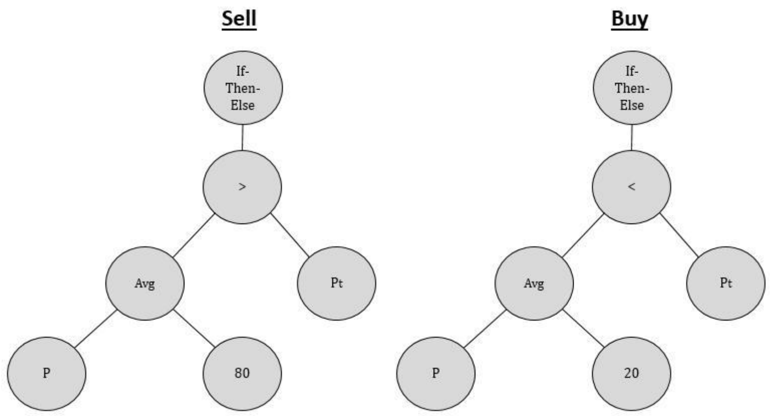

Figure 2. Performance through time: diversifying across asset

This figure reports the Fama-French four-factor alphas through time. The results are for a 5-5-1 case. Primary set is a randomly generated set of technical trading rules. Final Set is the set of optimized trading rules after applying optimization procedure. Moving-Average Set is the set of moving average strategies using lags of 3 to 100 days. The alpha values are computed on all of the NYSE/AMEX volatility decile portfolios. We average over the ten volatility decile portfolios and also the simulations.

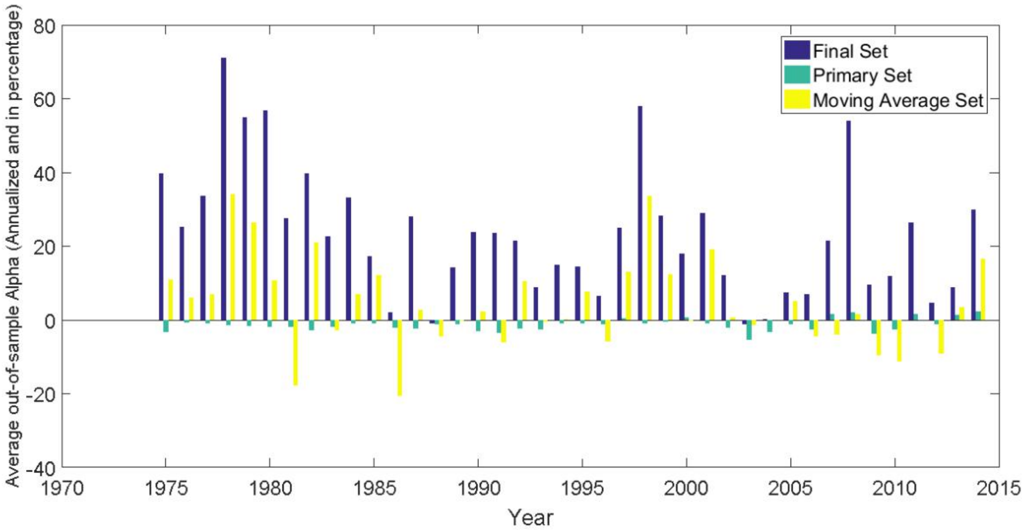

Figure 3: Annualized out-of-sample Sharpe ratio through time

This graph presents the average annualized Sharpe ratio over all the strategies across the volatility decile assets in out-of-sample periods. The setting is 5-5-1. We put all the optimized trading rules together and for each strategy we compute it's out of sample Sharpe ratio and next, we compute the average out-of-sample Sharpe ratios across rules and simulations for each out-of-sample year.

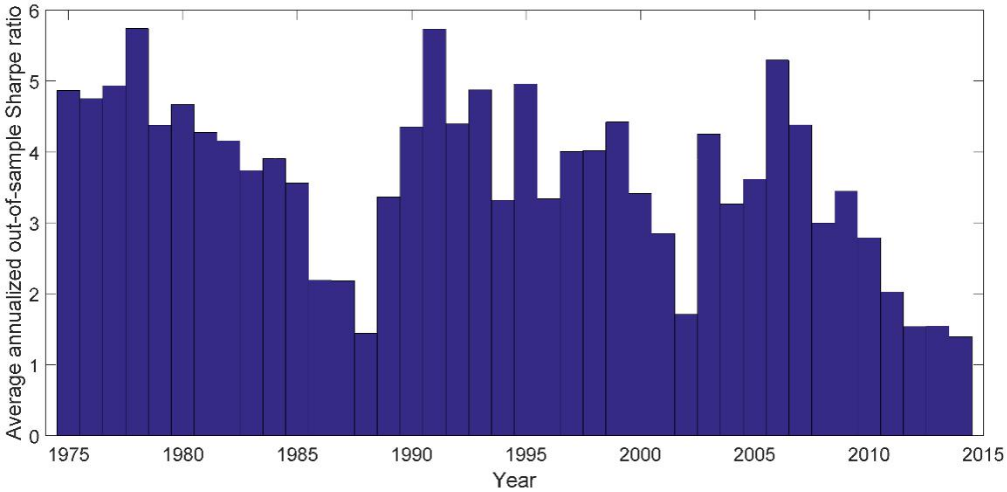

Figure 4. Transaction cost and number of trades through time

This Figure presents four-factor Fama-French annualized alphas through time with 5 bp transaction cost. This graph should four factor alphas for diversification across trading rules through time for the optimized trading rules (Final Set) after accounting for 5 bp transaction cost in returns for the trades. The alpha values are averaged over the trading rules, and the number of simulations.

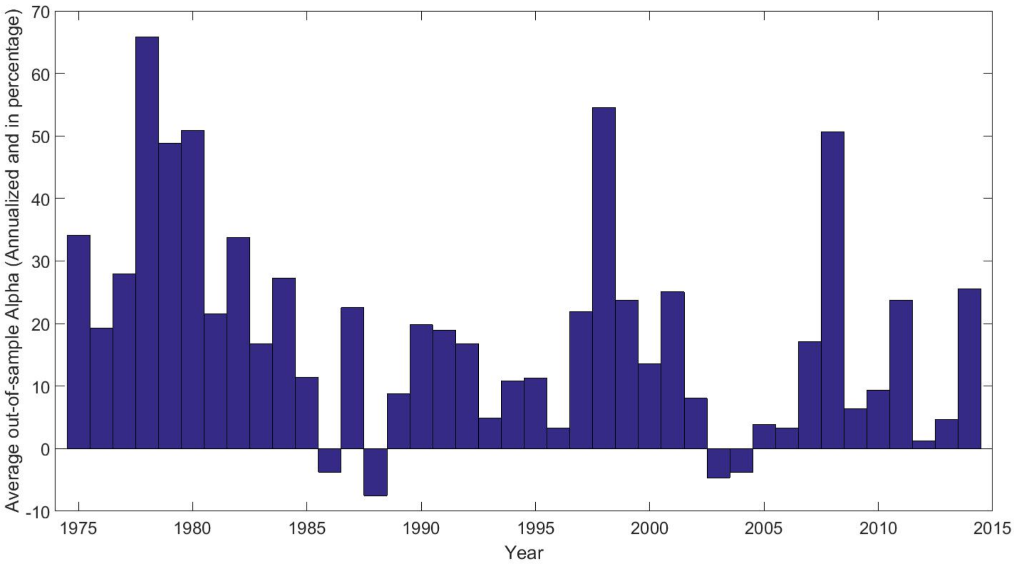

Figure 5: Trading Signals: Average number of trading signals through time
This graph shows the percentage of buy, sell and Hold risk free asset signals in the optimized trading rules in the Final Set through years in a 5-5-1 setting.

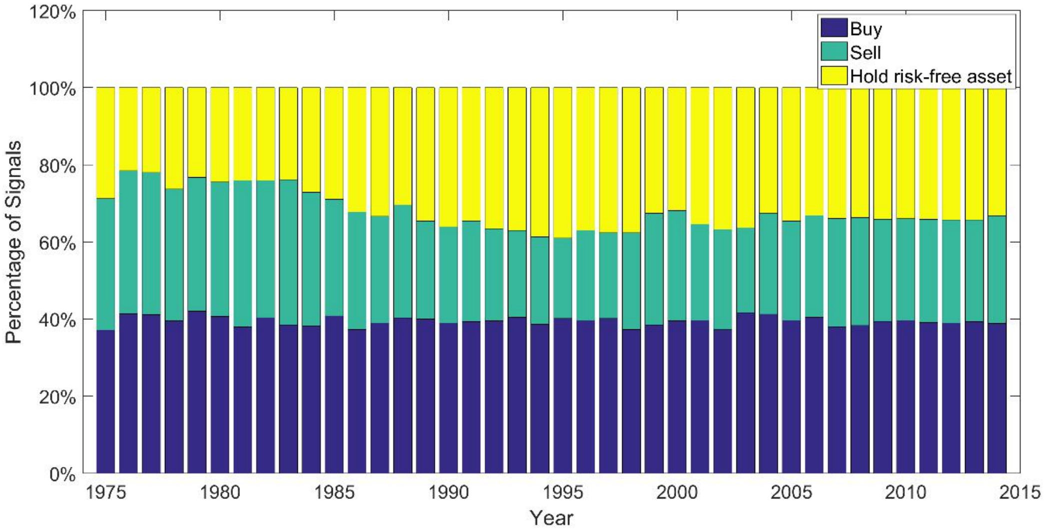

Figure 6. Average correlation across strategies through time

This figure shows the average out-of-sample return correlations among trading rules through time for all of the ten NYSE/AMEX volatility decile portfolios. The out-of-sample returns are computed for the optimized trading rules for the NYSE/AMEX volatility decile portfolios in a rolling fashion from January 1965. We adopt a 5-5-1 specification. We consider a five year training and a five year selection period in the algorithm.

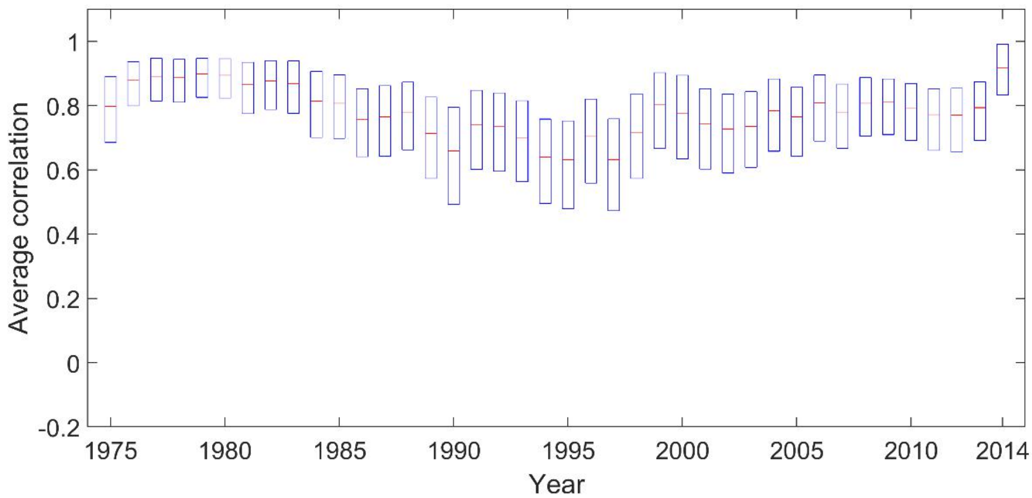

Figure 7. Weight allocation of volatility deciles through time

This table reports the weights invested in each volatility decile portfolio through time. We use a 5-5-1 setting. After running the algorithm on each decile portfolio, we find different number of trading rules for each volatility decile portfolio across simulations.

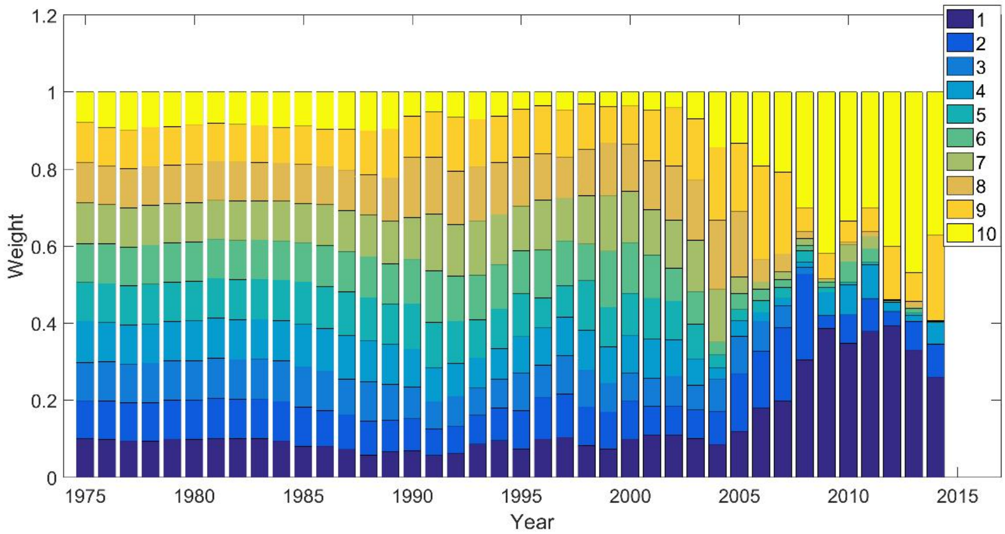

Table 1. Trading rule representation

This table presents the variables and functions employed in each level of a trading rule tree. There are four levels. In level 1, we consider Boolean operator and functions. In level 2, the relational operators. Real functions are assigned to level 3 where s stands for input variables. In level 4, we have two sets of inputs; the real variables and the terminals that specify the type of input variables.

Levels	Operator
Level 1:	
Boolean operator and functions	If-Then-Else
	and
	or
Level 2:	
Relational operator	<
	>
Level 3:	
Real Functions	Avg (s, days)
	Max (s, days)
	Min (s, days)
	Median (s, days)
	Lag (s, days)
	Volatility (s, days)
	RSI (s, days)
	Filter(s, Pr, days)
Level 4:	
Real variables	P : Price
	R: Return
	Pr: Random number in [-1, 1]
Terminals	Days: Number of days before today
	P_t: Price of the current day
	R_t: Return of the current day

Table 2. The searching mechanism

This table presents the algorithm designed to search for time-series anomalies. POP| is the size of the population and |GEN | is the number of generations that the algorithm is run.

Step 1
Generate a random population with |POP| individual trading rules (**Primary Set**)
Compute the fitness value of rules in the *training period*
Save the set of rules as the *Current Population*

Step 2
Compute the fitness value of the rules in the *Current Population* in the *selection period*
Save them as the initial best rules in the **Final Set**

Step 3
Crossover
Select two random rules from the *Current Population* at random and apply crossover operator
(Generate |POP| new rules)
Mutation
Select one random rule from the *Current Population* and also generate a new rule; Apply crossover operator
(Generate |POP|/2 rules)

Generate random individual rules (Generate |POP|/2 rules)
Calculate the fitness of the *new rules* in the *training period* and add them to the *Current Population*. Sort all the trading rules based on the fitness values in the *training period* and select the |POP| fittest rules as the *Current Population*

Step 4
Compute the fitness value of the new rules in the *Current Population* in Step 3 in the *selection period*
Add them to the best rules in the Final Set
Sort all the trading rules in the Final Set based on the fitness values in the *selection period* and select the |POP| fittest rules as the new best rules in the Final Set

Stop if |GEN| generations has passed
Back to step 3

Table 3. Summary statistics

This table reports the Out-of-Sample Fama and French four-factor annualized alphas in percentage. Primary set is a randomly generated set of technical trading rules. Final Set is the set of optimized trading rules after applying optimization procedure. Moving-Average Set is the set of moving average strategies using lags of 3 to 100 days. The alpha values are computed on all of the NYSE/AMEX volatility decile portfolios. The results are for a rolling 5-5-1 setting. The alpha values are averaged over the simulations and also the years for the Final and Primary sets and across the years for the Moving-average set. N is the number of unique technical trading rules. Specifically, for the Final Set, N is the average number of unique trading rules found in each rolling period and each simulation.

	Sets	N	Mean	SD	Skew	Kurt	25%	Median	75%
Low Volatility	Moving-Average	98	4.76	14.12	1.44	5.82	-3.08	2.15	12.53
	Primary	500	-1.26	13.97	0.43	4.86	-13.18	-5.32	2.35
	Final	332	14.27	15.21	1.14	4.27	2.28	9.26	23.17
2	Moving-Average	98	2.49	13.34	0.67	3.67	-5.17	2.22	12.08
	Primary	500	-1.34	13.68	0.08	4.83	-12.95	-4.77	2.13
	Final	304	16.52	15.39	0.91	3.17	4.23	11.33	22.36
3	Moving-Average	98	1.09	15.51	0.73	3.88	-7.59	0.68	11.32
	Primary	500	-1.25	13.79	-0.11	4.73	-12.62	-4.43	2.65
	Final	271	20.27	18.31	0.93	3.50	5.39	14.36	29.06
4	Moving-Average	98	0.33	18.83	0.10	5.60	-8.85	0.67	12.89
	Primary	500	-1.33	14.99	-0.26	5.08	-13.31	-4.62	3.07
	Final	285	23.86	19.64	0.65	3.13	7.20	19.71	34.90
5	Moving-Average	98	0.70	21.95	0.26	4.16	-9.94	0.81	14.97
	Primary	500	-1.12	16.06	-0.09	5.00	-13.33	-4.06	4.34
	Final	292	29.35	21.57	0.48	2.85	11.63	25.02	40.79
6	Moving-Average	98	1.86	23.79	0.27	3.56	-10.91	2.40	17.69
	Primary	500	-0.95	16.87	-0.11	5.14	-12.86	-3.42	5.42
	Final	302	29.82	20.92	0.33	3.10	14.10	27.38	41.11
7	Moving-Average	98	3.26	24.57	0.52	4.00	-11.39	2.70	19.63
	Primary	500	-0.82	18.19	-0.09	5.46	-12.88	-3.03	6.61
	Final	323	33.54	25.01	0.58	3.84	14.01	29.78	47.20
8	Moving-Average	98	7.79	25.79	0.54	3.76	-7.44	6.56	25.89
	Primary	500	-0.07	19.27	-0.15	4.95	-10.94	-0.01	10.94
	Final	338	33.33	22.95	0.59	3.85	16.85	30.91	43.69
9	Moving-Average	98	11.3	27.7	0.5	3.7	-4.7	9.5	30.4
	Primary	500	-0.3	22.3	0.1	4.7	-13.6	-0.9	11.6
	Final	361	32.4	27.4	0.4	3.0	10.7	30.7	47.5
High Volatility	Moving-Average	98	8.35	32.88	0.22	3.13	-10.08	8.65	30.61
	Primary	500	-4.92	37.68	-0.48	4.36	-37.95	-15.56	4.50
	Final	304	22.98	36.95	0.29	3.13	-3.22	20.46	44.51

Table 4. Summary statistics – Diversifying across assets

This table reports the Fama and French four-factor annualized alphas in percentage. Primary set is a randomly generated set of technical trading rules. Final Set is the set of optimized trading rules after applying optimization procedure. Moving-Average Set is the set of moving average strategies using lags of 3 to 100 days. The alpha values are computed on all of the NYSE/AMEX volatility decile portfolios. The results are for a rolling 5-5-1 setting. We are averaging across the assets, over time and also the simulations.

Sets	Mean	SD	Skew	Kurt	25%	Median	75%
Moving-Average	4.2	23.1	0.6	4.7	-7.5	3.3	17.6
Primary	-1.3	20.4	-0.8	9.0	-14.3	-4.2	5.2
Final	25.8	24.5	0.6	4.1	7.1	22.1	38.8

Table 5. Regression of alpha and time trend

This table reports the estimation results when regressing average four-factor annualized alphas of trading strategies ($AvgAlpha^t$) in out-of-sample years. We use the NYSE/AMEX volatility decile portfolio data from 1965 to 2015. We consider a 5-5-1 setting. We run the algorithm by rolling the input data by one year until we reach out-of-sample year of 2014. We create a time variable starting from 1 in 1975. Next, we compute the average alphas per year over the simulations and trading rules and regress the values on the time variable.

	Intercept	$year_t$	R^2
Low Volatility	15.64	-0.16	0.016
	(3.35)	(-0.79)	
2	27.09	-0.70	0.300
	(6.64)	(-4.03)	
3	35.43	-1.11	0.531
	(8.90)	(-6.56)	
4	41.59	-1.27	0.550
	(9.49)	(-6.81)	
5	48.46	-1.41	0.530
	(9.59)	(-6.55)	
6	48.26	-1.31	0.480
	(9.26)	(-5.92)	
7	54.21	-1.43	0.408
	(8.21)	(-5.11)	
8	53.68	-1.35	0.275
	(9.09)	(-5.38)	
9	53.30	-1.18	0.275
	(7.30)	(-3.79)	
High Volatility	12.21	0.20	0.005
	(1.06)	(-0.42)	
Diversifying across assets	36.15	-0.63	0.177
	(6.98)	(-2.86)	

Table 6. Summary statistics based on trading signals

This table reports the out-of-sample summary statistics for optimized trading rules in the Final set. The out-of-sample returns are computed for the optimized trading rules for the NYSE/AMEX volatility decile portfolios. The setting is 5-5-1. The values are averages over the trading rules and simulations over time. By Long, we mean the number of times we are either buying the asset or already bought the asset. By Short, we mean the number of times we have a short position (either already have a short position, or we take the short position following a sell signal).

		N (weight)	% of N	Avg Ret	Avg Std	Avg Skew
		101.998		0.113	0.003	0.325
Low Volatility	Long	53.412	52%	0.168	0.003	-0.084
	Hold risk-free	27.215	27%	0.023	0.000	-0.129
	Short	21.371	21%	0.130	0.004	0.249
	All	103.132		0.123	0.004	0.293
2	Long	54.351	53%	0.174	0.004	-0.030
	Hold risk-free	25.474	25%	0.023	0.000	-0.151
	Short	23.307	23%	0.155	0.005	0.200
	All	102.884		0.123	0.004	0.295
3	Long	54.192	53%	0.174	0.004	-0.028
	Hold risk-free	25.402	25%	0.023	0.000	-0.152
	Short	23.290	23%	0.156	0.005	0.201
	All	102.679		0.124	0.004	0.267
4	Long	51.927	51%	0.176	0.004	-0.019
	Hold risk-free	26.251	26%	0.023	0.000	-0.183
	Short	24.501	24%	0.155	0.005	0.149
	All	102.583		0.125	0.004	0.270
5	Long	51.952	51%	0.177	0.004	-0.018
	Hold risk-free	26.177	26%	0.023	0.000	-0.181
	Short	24.454	24%	0.156	0.005	0.152
	All	102.814		0.126	0.004	0.271
6	Long	52.009	51%	0.178	0.004	-0.016
	Hold risk-free	26.143	26%	0.023	0.000	-0.185
	Short	24.663	24%	0.158	0.005	0.151
	All	102.799		0.127	0.004	0.269
7	Long	52.050	51%	0.179	0.004	-0.016
	Hold risk-free	26.085	26%	0.023	0.000	-0.181
	Short	24.664	24%	0.158	0.005	0.151
	All	102.762		0.126	0.004	0.266
8	Long	51.980	51%	0.177	0.004	-0.012
	Hold risk-free	26.169	26%	0.023	0.000	-0.182
	Short	24.613	24%	0.158	0.005	0.139
	All	110.459		0.235	0.008	0.750
9	Long	45.627	45%	0.316	0.009	1.113
	Hold risk-free	26.802	26%	0.023	0.000	-0.377
	Short	38.029	37%	0.315	0.010	-0.271
	All	93.472		0.245	0.009	2.548
High Volatility	Long	36.852	36%	0.485	0.012	2.105
	Hold risk-free	28.651	28%	0.023	0.000	-0.159
	Short	27.968	27%	0.197	0.008	0.295
	All	102.558		0.147	0.005	0.539
Diversifying across assets	Long	50.435	49%	0.219	0.005	0.290
	Hold risk-free	26.437	26%	0.023	0.000	-0.190
	Short	25.686	25%	0.176	0.006	0.132

Table 7. Characteristics of trading rules

This table reports the number of real functions and input variables type in the Final Set. We use the highest and lowest NYSE/AMEX volatility decile portfolios. The optimization procedure is a rolling fashion 5-5-1 specification. All the values are averaged over the rolling out-of-sample windows over the 20 simulations.

Panel A: Real Functions

		Lowest Volatility Decile			Highest Volatility Decile		
	Set	Buy	Sell	Total	Buy	Sell	Total
Average	Primary	93.10	93.65	186.75	93.33	94.43	187.75
	Final	139.03	164.23	303.25	147.38	129.58	276.95
Minimum	Primary	95.75	93.65	189.40	94.63	93.10	187.73
	Final	201.08	155.18	356.25	140.48	132.98	273.45
Maximum	Primary	92.45	95.18	187.63	92.55	92.03	184.58
	Final	78.40	53.05	131.45	84.95	55.70	140.65
Median	Primary	94.05	93.88	187.93	91.15	92.80	183.95
	Final	135.70	156.40	292.10	137.83	156.28	294.10
Lag	Primary	94.28	92.78	187.05	91.75	94.15	185.90
	Final	54.05	44.45	98.50	87.13	65.93	153.05
Volatility	Primary	94.23	93.50	187.73	95.60	92.43	188.03
	Final	66.00	34.78	100.78	73.73	39.00	112.73
RSI	Primary	92.65	95.73	188.38	96.88	93.83	190.70
	Final	153.10	132.35	285.45	148.85	136.50	285.35
Filter	Primary	94.33	93.93	188.25	94.70	92.90	187.60
	Final	130.10	104.03	234.13	120.23	98.98	219.20

Panel B: Real Variables

		Lowest Volatility Decile			Highest Volatility Decile		
	Set	Buy	Sell	Total	Buy	Sell	Total
Price	Primary	500.18	501.63	1001.80	503.83	498.08	1001.90
	Final	1192.73	1119.43	1310.35	1167.58	1046.73	1212.40
Return	Primary	250.65	250.65	501.30	246.75	247.58	494.33
	Final	515.55	477.30	491.55	523.55	513.85	543.08

Table 8: Return correlation distribution among the rules in Final Set

This table reports the summary statistics for out-of-sample return correlations among trading rules in the Final set. The out-of-sample returns are computed for the optimized trading rules for the NYSE/AMEX volatility decile portfolios in a rolling fashion from January 1965. We adopt a 5-5-1 specification. Final Set is the set of improved trading rules after applying the algorithm to each volatility decile portfolios. We use four-factor model as the objective function in the optimization procedure. We consider a five year training and a five year selection period in the algorithm. The correlation values are computed over years and simulations.

	Mean	SD	Skew	Kurt	25%	Median	75%
Low Volatility	0.73	0.14	-0.72	3.88	0.65	0.75	0.83
2	0.73	0.15	-0.84	4.23	0.65	0.75	0.84
3	0.69	0.18	-0.72	3.76	0.58	0.71	0.81
4	0.61	0.21	-0.52	3.87	0.49	0.60	0.75
5	0.65	0.21	-0.62	3.45	0.52	0.67	0.80
6	0.64	0.20	-0.54	3.43	0.52	0.65	0.78
7	0.66	0.19	-0.62	3.69	0.55	0.68	0.79
8	0.65	0.19	-0.71	3.90	0.54	0.66	0.78
9	0.69	0.18	-0.84	4.04	0.57	0.71	0.83
High Volatility	0.71	0.18	-1.03	4.32	0.60	0.74	0.84
Diversifying across assets	0.77	0.19	-1.17	4.87	0.61	0.75	0.86

Table 9. Summary statistics – Placebo test

This table reports the Out-of-Sample Fama and French four-factor annualized alphas in percentage. We generate a set of optimized trading rules after applying optimization procedure. We use the ten NYSE/AMEX volatility decile portfolio returns from January 1965 to December 2014. Before applying the optimization algorithm, we scramble the returns by assigning a random number to each time period and sort the data by the random number. The results are for a rolling 5-5-1 setting. The alpha values are averaged over the simulations and also the years for the Final sets. N is the average number of unique trading rules found in each rolling period and each simulation.

	N	Mean	SD	Skew	Kurt	25%	Median	75%
Low Volatility	333	-3.45	9.70	0.25	5.48	-9.28	-3.32	2.20
2	304	-6.34	17.57	0.45	4.08	-17.46	-8.19	4.67
3	271	-7.90	26.15	-0.13	3.20	-24.90	-8.09	9.83
4	285	-8.48	23.10	-0.53	3.44	-23.46	-6.02	7.87
5	292	-11.38	27.15	0.36	3.10	-30.12	-13.88	3.95
6	303	-10.16	34.01	0.83	5.01	-31.61	-11.82	6.64
7	323	-26.63	30.50	-0.28	3.27	-45.98	-26.52	-3.97
8	338	-22.66	41.98	-1.44	8.72	-39.61	-19.30	0.23
9	361	-9.96	34.50	0.38	3.27	-33.87	-11.41	11.64
High Volatility	305	-47.43	43.20	-0.54	3.93	-71.45	-43.97	-18.98

Table 10. Size decile portfolios

Panel A reports the Kendall rank correlation coefficients between the size deciles and the volatility decile portfolios. We use the ten NYSE/AMEX/NASDAQ size decile portfolios and the ten NYSE/AMEX volatility decile portfolios from January 1965 to December 2014. Panel B reports the average alphas for three sets of trading rules. Moving average set is the set of moving average strategies. The primary set is the set of randomly generated trading rules and the Final set is the set of optimized trading rules. The underlying setting is 5-5-1 and we pull all the out-of-sample alphas across years together and compute t-statistics across rules in the sets.

Panel A: Correlation matrix

		Size decile portfolios									
		Large	2	3	4	5	6	7	8	9	Small
Volatility decile portfolios	Low Volatility	0.31	0.38	0.41	0.43	0.44	0.46	0.47	0.48	0.50	0.49
	2	0.38	0.46	0.51	0.56	0.59	0.62	0.63	0.66	0.68	0.65
	3	0.40	0.48	0.54	0.59	0.63	0.66	0.69	0.71	0.74	0.70
	4	0.41	0.50	0.56	0.61	0.66	0.69	0.72	0.74	0.77	0.70
	5	0.42	0.51	0.57	0.63	0.68	0.72	0.74	0.77	0.78	0.70
	6	0.43	0.52	0.59	0.65	0.70	0.74	0.76	0.78	0.78	0.68
	7	0.45	0.54	0.61	0.67	0.72	0.75	0.77	0.78	0.78	0.66
	8	0.46	0.56	0.62	0.69	0.73	0.75	0.76	0.77	0.76	0.63
	9	0.49	0.58	0.64	0.69	0.72	0.73	0.73	0.72	0.70	0.58
	High Volatility	0.53	0.59	0.61	0.61	0.60	0.58	0.56	0.55	0.53	0.43

Table 10 Continued

Panel B: Alphas for size decile portfolios

	Sets	N	Mean	SD	Skew	Kurt	25%	Median	75%
	Moving-Average	98	-6.21	19.98	-0.64	6.74	-17.64	-5.05	6.39
Large	Primary	500	-0.45	13.57	0.07	8.52	-8.70	-1.50	4.72
	Final	162	19.60	16.25	0.24	2.48	4.77	17.53	32.05
	Moving-Average	98	1.49	21.19	0.50	4.00	-10.80	1.30	15.26
2	Primary	500	-0.46	16.85	-0.17	5.11	-11.04	-1.66	7.06
	Final	325	37.38	21.20	0.28	3.90	22.91	33.60	46.51
	Moving-Average	98	1.94	23.32	0.33	3.91	-10.76	1.39	16.52
3	Primary	500	-0.21	16.75	-0.12	4.80	-9.96	-0.68	8.14
	Final	335	35.95	21.92	-0.05	3.41	21.75	35.92	48.79
	Moving-Average	98	3.68	24.03	0.38	3.80	-8.84	3.72	19.95
4	Primary	500	-0.04	16.71	-0.05	5.20	-9.42	-0.11	8.63
	Final	328	34.82	23.11	0.15	3.31	17.91	32.62	47.02
	Moving-Average	98	5.77	23.50	0.48	4.42	-6.95	6.82	20.34
5	Primary	500	0.05	16.57	-0.02	5.08	-8.74	0.39	9.20
	Final	333	31.58	23.43	0.31	3.61	15.57	29.93	42.87
	Moving-Average	98	9.27	23.72	0.61	4.31	-5.45	8.97	21.91
6	Primary	500	0.27	17.82	0.14	5.00	-8.75	1.00	11.03
	Final	326	28.90	24.88	0.46	3.00	7.39	25.87	42.26
	Moving-Average	98	14.43	25.16	0.67	3.84	-2.07	12.32	30.26
7	Primary	500	0.53	20.80	0.19	5.17	-9.21	2.21	13.76
	Final	373	30.63	24.96	0.21	2.71	10.72	28.28	45.98
	Moving-Average	98	19.01	26.40	1.48	6.75	2.50	15.43	31.37
8	Primary	500	0.79	23.61	0.32	6.95	-8.76	3.18	15.71
	Final	407	32.12	27.21	1.48	6.00	12.74	24.83	44.43
	Moving-Average	98	24.6	28.3	1.7	8.9	5.5	21.9	38.4
9	Primary	500	1.1	27.7	0.4	7.7	-9.0	4.4	19.3
	Final	412	34.6	28.5	1.6	7.5	14.7	30.1	47.0
	Moving-Average	98	25.78	33.44	0.62	5.65	6.20	19.04	38.58
Small	Primary	500	1.12	30.12	0.20	5.72	-11.36	5.79	20.70
	Final	370	31.03	30.07	0.97	5.80	12.38	27.14	46.02
	Moving-Average		9.97	26.73	0.87	6.26	-6.03	7.81	24.52
Diversifying across assets	Primary		0.27	20.80	0.39	8.05	-9.43	0.70	11.51
	Final		32.32	25.44	0.83	5.35	13.81	29.44	45.25

Table 11. Other factor models - Out-of-sample Alpha Distribution

This table reports the summary statistics for out-of-sample annualized alphas in percentage for three asset pricing models: CAPM in Panel A, Fama and French three factor model in Panel B and Fama and French five factor model in Panel C. The underlying assets are NYSE/AMEX volatility decile portfolios. Moving-Average Set is the set of moving average strategies using lags of 3 to 100 days. Primary Set is the set of trading rules generated randomly following the routine in Section 2. Final Set is the set of improved trading rules after applying the algorithm. We consider a five year training and a five year selection period in the algorithm starting. The setting is 5-5-1. The average alphas come from diversification across all the volatility decile portfolios.

Panel A: CAPM

Sets	Mean	SD	Skew	Kurt	25%	Median	75%
Moving-Average	7.28	23.96	1.25	6.72	-5.75	4.23	19.42
Primary	-1.19	22.57	-0.32	7.40	-15.71	-4.00	7.24
Final	23.59	25.46	1.07	5.60	6.24	20.23	35.15

Panel B: Three Factor

Sets	Mean	SD	Skew	Kurt	25%	Median	75%
Moving-Average	4.96	22.43	0.58	4.74	-7.11	3.94	18.47
Primary	-1.08	20.17	-0.74	8.95	-13.47	-3.39	6.07
Final	25.09	23.43	0.54	4.14	7.14	21.84	37.94

Panel C: Five Factor

Sets	Mean	SD	Skew	Kurt	25%	Median	75%
Moving-Average	7.33	33.67	1.50	7.06	-10.93	-0.16	19.74
Primary	-1.41	13.64	0.24	4.24	-13.41	-5.49	2.35
Final	13.08	13.51	0.89	3.52	1.89	9.19	20.32

Appendix

A. Crossover Operation

The goal of the crossover operation in genetic algorithm is to generate two new offsprings with shared characteristics of their parents. We explain the procedure with an example. Suppose we randomly select two trading rules from the existing population labeled as Parent 1 and Parent 2. An example is presented in Figure A.1. In the cross-over procedure we substitute the left branch in the Buy-side of Parent 1 with the left branch of Sell-side of Parent 2. The substitution generates two new offsprings added to the existing population. The offsprings are illustrated in Figure A.2.

Figure. A.1. Sample trading rules

The Trading rules are chosen randomly from the existing population for the purpose of crossover operation. We copy the buy-side branch of Parent 1 and copy it to the sell-side branch of Parent 2; next, we copy the sell-side branch of Parent 2 to the buy-side of Parent 1.

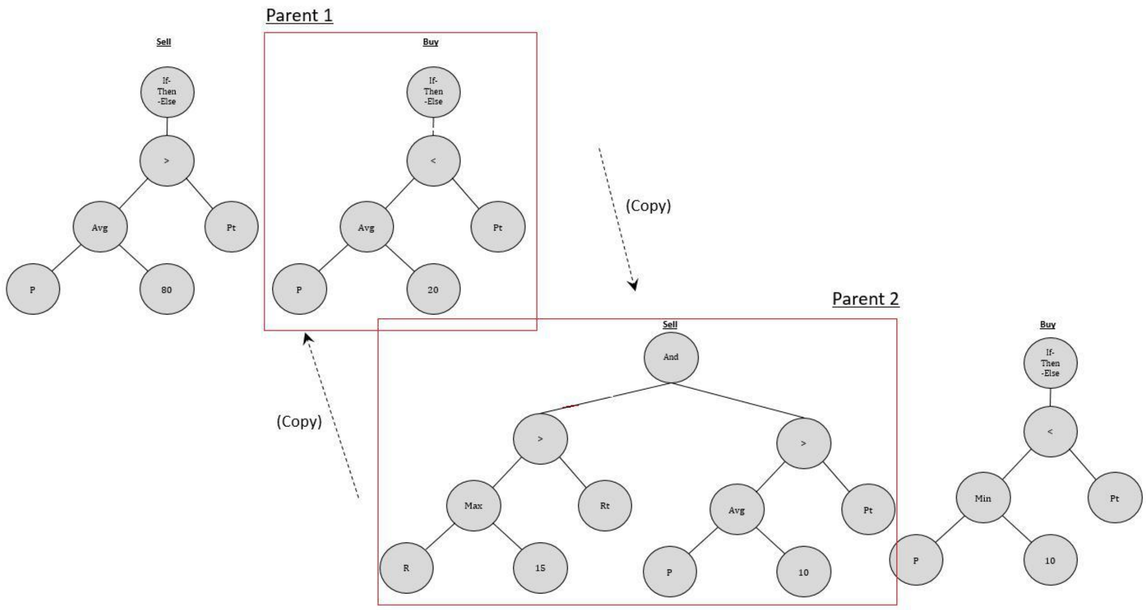

Figure. A.2. Trading rules as a result of crossover operation
This figure presents the resulting trading rules (labeled as offsprings) from crossover operation on the trading rules (Parents) in Figure A.1.