Moving Average Learning Strategies for Technical Analysis: 
Implications for the Efficient Market Hypothesis
by Daniel Svogun

Abstract: Technical analysis seeks to find patterns in past stock prices that are profitable to trade on in the future. By restricting the Allen and Karjalainen genetic algorithm to find parameters of a restrictive technical analysis rule type, I demonstrate eight time deciles of technical analysis profit beyond realistic transaction costs over the last twenty years in the component DJIA stocks over the 5-minute time frequency, and nine to ten time deciles in the ETF DIA (a DJIA tracking ETF) over the 5-minute time frequency and the MA parameter, depending on the summary statistic. These results are largely robust to slippage. As a whole, the results are on the margin of violating the Efficient Market Hypothesis (EMH) but ultimately confirm a weak form of the EMH.

Introduction:

The efficient market hypothesis states that any systematic way to profit beyond a buy and hold strategy will be incorporated quickly into the market such that the system is no longer better than buy and hold. This hasn’t stopped technical analysts or “chartists,” as they were once known, from looking for opportunities to beat buy and hold. Starting with Charles Dow, the founder of the Wall Street Journal and co-founder of the Dow Jones Industrial Average, chartists have looked for patterns in stock price charts over time, and created rules that they believed would beat buy and hold, based on the shapes that appear in those charts. As this method is primarily intuitive and un-systematic, economists, such as Lo, Mamaysky, and Wang, hereafter LMW, would later take the most popular intuitive technical analysis patterns, and make them systematic and mathematical. LMW’s purpose was only to find whether these patterns contained information, and by comparing conditional and unconditional distributions of returns before and after patterns were observed, they determined that seven of the ten patterns they tested contained information (LMW 2000).

The exact nature of these systematic tests of technical analysis vary dramatically. In this dissertation, I focus on a learning algorithm in a genetic algorithm framework. Genetic algorithms, as the name suggests, are modeled after the process of natural selection found in nature. When applied to technical analysis, as they are in the Allen and Karjalainen paper, hereafter AK, they start out with randomly created buy and sell rules. These rules are tested for their profitability over a training period, say three days of 5-minute frequency prices. The most profitable rules are then combined with each other, mimicking the mating of the most survived organisms in nature. The “children” of these rules are then tested again for their profitability and recombined, in an iterative process always biased towards the most profitable rule, for some set number of generations, typically 100. After 100 generations, the most profitable rule or rules are selected to trade on in the trading period, say the next three-day period. The profit is calculated, and it is compared to the returns had the individual simply bought at the beginning of the trade period and sold at the end, also known as “buy and hold (AK 1999).”

Genetic algorithms practically eliminate the issue of data snooping. In data-snooping a researcher might, advertently or inadvertently, learn that a certain specific trade rule was
very popular over some period and use it. Even if (s)he finds it profitable over this period, it doesn’t say anything about that rule’s viability unless a trader could reasonably have known about it before the period began. Because genetic algorithms “teach” themselves to trade only over a set training period, that occurs before the trading period, any rule it comes up with is, by design, available to researchers and traders before-hand.

In this paper, I design a learning algorithm in a genetic algorithm framework. Typical genetic algorithms randomly create their “seed” rules before recombining them, and so there is usually a difference between the returns and rules that result from a larger number of seeds. In fact, every time the algorithm is run, the results are likely to be different. To be clear, the returns and rules do tend to asymptote and converge after some number of generations, but there is still this randomness implicit in the genetic algorithm’s design. AK acknowledge this problem in their appendix. On the other hand, while Hsu and Kuan (2005) learning strategies do adjust for previous profitability and have a momentum measure, the comparison values used are set, based on those that they were told are profitable, and so subject to the data-snooping problem. Instead I take the advantages of both, while eliminating weaknesses. By limiting the genetic algorithm to a single parameter measure of momentum (here, the simple moving average) as a trade rule, and taking only the values that occur in the training period as comparison values, I create a learning algorithm for technical analysis within a genetic algorithm framework.

Past literature, cited in depth below, has found significant variance in returns based on several parameterizations of genetic algorithms and I test many of these here. All returns are measured using the standard in the literature, the BETC, or break-even transaction costs, that is the transaction cost at which the trading strategy and buy-hold return the same amount. If the BETC is greater than realistic transaction costs, the strategy is said to violate the Efficient Market Hypothesis (EMH). The first variation I test is based on the time-period at which prices are taken. Becker and Seshradi, for example, finds profitability on monthly prices, while many don’t on a daily price basis. I look at prices on a daily and 5-minute frequency. I also distinguish between the profitability of an index (here, the Dow Jones Industrial Average), using a close-tracking Exchange Traded Fund (ETF), the DIA, and the average of its component stocks. Next, I look at the difference in returns when including a transaction cost within the training period, versus not, as well as differing ways to compare the returns of the different algorithmic parameterizations I look at here. After these tests, it becomes clear the highest BETCs occur in the Moving Average with Transaction Costs in training at the 5-minute price frequency. I find eight periods out of ten of greater than realistic transaction cost BETCs in the Dow Jones Industrial Average component stocks. I find nine periods out of ten of greater than realistic transaction cost BETCs in the ETF DIA, when using the mean summary statistic. The median trade rule of the ETF DIA in 5-minute time period is found to violate the EMH in all ten deciles of time, a marginal EMH violation.

I robustness test the BETCs this parameterization achieves in two ways, both of which are contributions of the paper. In the first, I consider a case of “slippage,” that is, at higher frequency trading, sometimes the algorithm desires to purchase at some price, but that price exists for such a short period, that it is not completed. To model this, the algorithm randomly misses 25% of the trades it wants to complete, and I run 20 simulations of these cases. This means that the mean BETC of these 20 simulations are expected values and can be compared to the assured transaction case. Next, the algorithm
design implicitly assumes that a trader could purchase or sell a stock at the price of the most recent transaction. This is not technically true. Instead in the second robustness test, I test a period of 5-days in which the algorithm can only buy at the prices sellers want to sell at and only sell at the prices buyers want to buy at (using bid-ask data). I then compare the BETCs of this period with and without this requirement. I find my results largely robust in both cases. I conclude that while past paper standards would indicate these results to be violations of the EMH, the higher standard used here, requiring full, all-ten time decile persistence does not. While these results are about as close to a full EMH violation as possible without violating it, they still suggest that a weak efficient market hypothesis holds.

Literature Review:

Since the skepticism that prevailed for many years about “charting” and particularly the last two to three decades, significant progress has been made demonstrating the usefulness of technical analysis. Wilcox and Crittenden find using 24,000 stocks over 22 years the profitability of technical analysis. Faber 2007 shows that technical analysis improves risk-adjusted return. LeBaron 1999 and Neely 2002 show substantial gains using the MA signal, while Gehrig and Menkhoff 2006 argue that technical analysis is as important to currency traders as fundamental analysis is. Many argue that incomplete information on fundamentals is the key aspect that leads traders to use technical analysis. Daniel, Hirshleifer, and Subrahmanyam in 1998, and Harrison and Stein in 1999 show behavior biases that lead to price continuation.

As the Bajgrowicz and Scailler, hereafter BS, evidence implies mature markets, like the DJIA, tend not to have profitable technical analysis trading rules, particularly during more recent time periods (BS 2012). Hsu and Kuan, hereafter HK, test several technical analysis trading rules over several different markets (some more mature than others). They also make a key observation – technical analysts don’t just make one simple rule and follow it blindly. Quoting Pring (1993), they say “No single indicator can ever be expected to signal all trend reversals, and so it is essential to use a number of them together to build up a consensus.” In other words, typically, they take information from a lot of sources and try to discern the most likely future from it. To the end of modeling this correctly, they develop “investor’s strategies” that allow investors to combine buy and sell signals from simple rules in different ways and test their outcome (HK 2005). Like BS, their simple rules are often parameterized using rules in previous papers such as LMW, Edwards and Magee, and Chang and Osler. They also develop learning strategies, which allow investors to test different measures of performance and then change their simple strategy based on the one that is best performing by these varying measures. Over the four different markets, they only find profitable rules in the Nasdaq Composite and the Russell 2000, but none in the Dow Jones Industrial Average and the S and P 500. Additionally, even while allowing for these investor strategies, they find no profitable strategies from 1990 to 2000. Both of these results support the general trend from BS and others, that more mature markets and later time periods imply less profitability of technical analysis.

HK discuss the proper transaction fees, and they determine a fee of 0.05% - 0.13% (or 5-13 BPs) as the minimum one-way trade. They use 39,832 rules, 12 classes of 18,326 simple rules, contrarian rules and 3180 investor’s learning strategies. In
the markets in which Technical Analysis is profitable, and even in many that it isn’t, they find that learning strategies, and 2 and 5-day Moving average strategies to work quite well. In the Russell 2000, a learning Moving Average strategy performs the best. Additionally, and quite interestingly, learning strategies that are profitable often have simple strategy components, none of which are profitable by themselves. The clear implication is that technical analysis cannot be evaluated based solely on the results of simple rules. Of the top ten most profitable strategies in the Nasdaq Composite and the Russell 2000, eighteen of the twenty were either a Moving Average strategy or a learning strategy, and many of them were a learning strategy within the moving average strategy class. It is important to note that the best rules do not uniformly beat buy and hold in all periods studied, but they still “compare favorably” to buy and hold, both in and out of sample periods. They conclude that these results support Siegel (2002), that is that weak market efficiency have not yet formed in “young” markets.

The papers above have demonstrated that learning strategies that select between or combine simple strategies rather than the set simple strategies improves the performance of any single strategy. Even when these strategies work with reasonable transaction costs assumed, there is often a problem of data snooping, that is cherry picking only the most profitable simple strategies for a past period and implicitly and often erroneously asserting that traders at the time would be able to choose these strategies. Several of the papers cited here, such as BS, ground their contribution in tests that they say determine whether traders could have reasonably chosen the tests beforehand, specifically the FDR and White’s Reality Test. Papers which present one, will often present problems of the other, and assert theirs is better. This paper is agnostic on the proper data snooping statistical test front, rather it takes an approach that by design avoids the data snooping problem. In their seminal paper “Using Genetic Algorithms to find technical trading rules,” Franklin Allen and Risto Karjalainen, hereafter “AK,” use genetic algorithms to select technical analysis trade rules. The genetic algorithms they develop work by splitting the data set into training and trading periods, in which the algorithm is trained for a set period and then trades with what it has learned over the next period. In this way, the algorithm only uses information that was available before the trading period, and so, by design, avoids the data snooping problem.

Genetic algorithms are a subset of evolutionary algorithms, themselves a subset of machine learning techniques. In a genetic algorithm, the researcher starts with a population of randomly generated solution candidates. In an application of technical analysis, these might be if moving average is greater than some number, buy, or sell, and so on the with different categories of technical analysis rules like loss-prevention rules, ceiling rules, candle-stick comparison et cetera. Next, the algorithm randomly selects pairs of “parent” rules to recombine, with this random process biased in favor of the rules that are relatively “fit.” Here, a fit rule is often a simple operator that chooses the rules with highest returns over the training period, but the framework they build allows for almost any arbitrary rule or function. Next, some cross-over operator determines how the rules are combined, often with some small chance of a mutation in order to maintain genetic diversity. Finally, these new rules are tested on the training period. This testing and recombination occur iteratively for either some x number of generations or until the rules no longer improve beyond some set rate. A footnote emphasizes that the simple genetic algorithm framework they present is quite malleable and offer that there are varying ways
to handle each generation, including some that eliminate the entire previous generation. Other adjustments include algorithms that penalize evolved solution candidates for complexity or have a cap on it. Not only do genetic algorithms allow a technical analysis researcher to work on often untenable or difficult problems, such as non-differentiable and discontinuous objective function maximization but it is also less likely to converge to a local maximum than hill-climbing or gradient-type methods. There are a few additional computational quirks of genetic algorithms that are important to note. While they "lead to a favorable tradeoff between exploitation of promising directions of the search space and exploration of less-frequented regions of space" there is no guarantee that the genetic algorithm will converge to the global optimum. However, studies on genetic algorithms show that with more generational iterations that the probability of convergence to the global optimum tends to asymptote to 100%, a process that can be improved by strategically switching to hill-climbing techniques. Still, it is important to note that in the form described by AK, the genetic algorithm is not technically a deterministic process, and that AK tested running the same algorithms on a more computationally powerful setup with more generations and a larger starting population of solution candidates and got better results. They discuss this in their paper’s appendix.

Genetic algorithms have been used in economic markets, time series forecasting and econometric estimation, while string-based genetic algorithms have been applied to finding market-timing strategies based on fundamental data for stock and bond markets by Bauer (1994). AK apply their genetic algorithm to learning technical rules in the S and P 500 using daily prices from 1928 to 1995. They run their algorithm using .1%, .5% and .25% transaction costs per one-way trade. When including these trading costs, the rules do not earn consistent excess returns over a buy and hold strategy in the out-of-sample periods. However, it is able to identify and enter the market to take advantage of periods of positive daily returns. In other words, ignoring transaction costs, most trades do result in a positive profit. AK assert these results are largely explained by the well-known phenomena of low-order serial correlation in stock index returns. They test this by delaying trade signals by one day. In doing so, the trading signals lose most of their forecasting ability, “indicating that the rules exploit positive low-order serial correlation in stock index returns.”

Additionally, like spreading an investment among several vehicles has a positive diversification effect, there appears to be a diversification benefit in using a number of trading rules in a portfolio, with the position to be taken in each period determined as a weighted average of the trading rule signals. Finally, they acknowledge, that like most technical analysis models, theirs ignores dividend returns, and so will underestimate buy and hold return, and to a lesser extent underestimate the returns on the technical analysis strategies. This only strengthens their results as it would tend to increase the buy and hold returns relative to any trading rules.

As far as rule type, the most successful strategies are often based on momentum, and even those that aren’t, often have mathematical forms that are quite similar. Schwert 2003 goes so far as to say that momentum anomaly from JT 1993 is the only one that is persistent and has survived since its publication. A 2012 paper by Han, Yang and Zhou explore the momentum phenomena and find that two common ways of measuring momentum seem to measure separate phenomena. They find that portfolios sorted into deciles ranked by volatility generate abnormal returns, even when risk-adjusted with the Capital Asset Pricing Model (CAPM) and the 3-factor Fama-French model. They find similar
results hold when these portfolios are ranked based on other proxies of information uncertainty. They aren’t alone in this assertion, as Neely, Rapach, Tu, and Zhou show that technical indicators are at least as good as popular macroeconomic data when forecasting the stock market, while Goh, Jiang, Tu, and Zhou show that technical analysis works much better than these macroeconomic data points to forecast bond markets. They posit that if technical signals are truly profitable, then they are more likely to show up in low-certainty vs. high-certainty situations. That is, in general, the more noise-to-signal, or the more uncertain the information, the more profitable technical analysis. In addition to this decile splitting model, a Han, Yang, Zhou working paper also apply an MA investment timing strategy to each decile. For a given Moving Average Portfolio (MAP), the strategy is to buy or continue to hold the portfolio today when yesterday’s price is above its 10-day Moving Average price. If this is not true, then the strategy invests in the risk-free asset, here the 30-day T-bill. While a number of papers above have demonstrated the profitability of technical analysis when assuming zero or quite small transaction costs, the MAP process described in HYZ allows for break-even transaction costs that are “reasonably high,” suggesting that corresponding MA timing strategies should still earn economically significant returns after the appropriate transaction costs. They, like others, use the term “break even transaction costs” or BETC to quantify the phenomena. As the name suggests, it is the transaction cost at which the technical analysis trade rule with a transaction cost per buy or sell decision and buy and hold over the same period result in the same return. In some cases the BETC is over 50 basis points, higher than what most economists consider reasonable for the time period tested.

By tweaking the very general AK Genetic Algorithm and Programming framework, Becker and Seshradi, in their 2003 paper “GP Evolved Technical Trading Rules can Outperform Buy and Hold” find that mean returns with their technical trading algorithm beats buy and hold 99.5% of the time with a substantial transaction cost. One of these modifications, crucially, is using monthly rather than daily data. They contend that their results call into question the Efficient Market Hypothesis in “even its weakest form.”

In a 2000 paper, Dempster and Jones again adjust the AK framework to positive effect, this time for the foreign exchange, FX hereafter, market. In the FX market technical analysis is a much more developed field, so it makes sense that an equity technical analyst may find some “upstream” ideas in the current technical analysis FX protocols. In their surveys, Taylor and Allen (1990-1992) found that over 90% of surveyed London FX dealers and traders use technical analysis of some sort. Lui and Mole found that technical analysis is significantly more popular than fundamental analysis in shorter time horizons. Most successful traders in FX don’t use one technical analysis trade rule blindly, but rather switch between them based on some kind of feedback system, typically regarding the recent return earned with it. They notice that most researchers focus on the trading rules, while relatively little attention is paid to the system construction. Also, unlike the previous cited technical analysis papers, they use intraday data. Their protocol, uniquely, has an option to adjust for slippage, that is the amount lost when trying to aim for a price on a small-time scale that is missed. This becomes an issue when dealing on minute or second timescales and the algorithm is attempting to earn an additional return from micro-fluctuations in price. They construct a genetic algorithm that buys and sells on a 15-minute frequency time-scale overlaid with a loss-mitigation function that operates on a 1-minute frequency time-scale. Their genetic algorithm includes the MA crossover indicator as well
as several others commonly used. They find that a majority of these adaptive entry models lead to an outcome worse than just buy and hold, however here it is possible to profit from trading technical rules when the trade entry signal is taken from combining indicators and they note that this is how a trader would do so. Adaptively using only the best strategy yields 7.4%, or 40 basis points better than static equivalent. While they find their 15-minute strategy's results are promising enough for future work (their stated goal of the research), they describe the daily time frequency returns over the same period as “disappointing.”

Data Description:

The stock price data, which measures only the price at which transactions are completed, is compiled from two sources. The first is The Center for Research in Security Prices, or CRSP. This is where I get the daily stock price for all of the historical stocks that have been a part of the DJIA. The five-minute frequency stock prices of the DJIA comes from the online data-source Kibot, which provides this data for a modest fee. In both cases, these data have been adjusted for stock-splits. It does not include any data on dividends. The project uses all available and applicable data at each of these data providers. In the case of daily data, the starting time varied dramatically, with data from quintessential stocks like Coca-Cola starting as early as 1925, while some later companies only join the DJIA in the last decade or two. The 5-minute data was more consistent in timing, with the majority of it starting around 1998 and continuing until Spring 2018, roughly 20 years of data. In cases where there was not enough data available to run the entirety of the algorithm, such as when a company was listed very briefly on the DJIA, the company was eliminated from the analysis. The price at any given day or time used in this project is the “close price” at that time. The 5m Bid-Ask close data was from Kibot, while the second-by-second Bid and Ask data was provided by my dissertation advisor, Professor Rengifo.

Methodology

The papers in the previous section point to a few salient facts about technical analysis. When applying simple technical analysis rules over large periods of time, they tend to be more profitable the earlier the time period tested, and the less mature the market. Possible explanations for these include a higher transaction cost and therefore larger unexploited technical analysis patterns in the past, and markets becoming more efficient as they mature, respectively. When technical analysis is found successful more recently and in more mature markets, there is some kind of systematic and often dynamic way in which simple rules are applied. These range from the decile sorting of MAPs, to learning/voting algorithms, to genetic programming and algorithms, with other variations possible. As demonstrated in Dempster and Jones (2000), there can be a large difference in cumulative returns net transaction costs over the frequency at which the rules are allowed to trade, even over the same asset. To be fair, they studied FX markets, but it is reasonable to believe this could happen also in equity markets.

There are three additional components that are important to recognize. First, there is the transaction costs. Any time a technical analyst buys or sells an asset, (s)he will have some marginal cost associated with it, and many papers incorporate these. The literature
standard is to measure these costs in basis points, or hundredths of a percent. In the last several decades, many brokers, particularly those that cater to traders who buy and sell many equities per minute, have moved to a flat rate price per trade. The standard price is around half a cent per order (buy or sell) and, with volume discounts, they are often significantly lower (Reinkensmeyer). It should be noted that these prices include all regulatory fees on stock transactions, meaning the marginal cost alone of executing a trade is surely lower. Importantly, these transaction fees have decreased over time. A transaction that may have taken hours or even days to complete a century or more ago, with high costs associated, now takes place in the blink of an eye electronically, with very little marginal costs associated. Although it is by no means universal, some exchanges in the United States provide a very small refund for every high frequency buy and sell order, made in the hundreds of thousands or millions of orders by firms that specialize in it. The indices see this as a way to increase the liquidity of the assets included. Economically speaking, this means there is a negative transaction cost. Secondly, the speed at which transactions can occur is ever increasing. The smallest time-frame considered in the papers cited above was a one-minute frequency, but current technology allows traders to trade on a nano-second frequency, with pico-second frequency in the works. Minute-by-minute prices, for example, smooth over the volatility between them that occur in second-by-second prices, and so on.

Finally, the bar with which economists measure a successful trade rule varies dramatically from paper to paper. The seminal paper in technical analysis had a goal of proving the information content of technical analysis rules, by looking at distributions of returns chosen by a given rule versus those not (LMW). Later papers had a goal of simply proving that some kind of technical analysis rule was profitable to use at some point. The lowest possible returns comparison bar used was the comparison of the trade rule returns with the risk-free asset. The most commonly used bar is the comparison of the trade rules with buy and hold, that is the return of the asset had the investor bought the asset and held it for the period over which the technical analysis trade rule is run. Finally, many papers after this add a transaction cost per trade when comparing the returns to buy and hold. These transaction costs are almost always measured in basis points (BP), or hundredths of a percent of the price of the stock, and range from 5 BP on the low end to about 50 on the high end. The consensus of the papers above finds certainly that there was a period tested in the BLL paper in which technical analysis was profitable over reasonable transaction costs and buy and hold, but the Ready (2002) paper shows quite persuasively that economists ought to treat this period as an interesting anomaly of inefficiency rather than apply any systematic significance to it. The AK genetic algorithm and several specific algorithms in the AK framework typically beat buy and hold without transaction costs, while a few with specific tweaks beat the buy and hold benchmark even with reasonable transaction costs. HYZ (2012) do show MAPs with MA timing strategies applied on top of them to be quite profitable, even with the addition of transaction costs of 50 basis points in some cases, but because they only report the results of a simple trade rule, the data snooping problem remains an issue. They admit in footnotes to testing other simple trade rules on the data that do not do as well.

The AK genetic program/algorithm is a remarkable tool because it allows the researcher to eliminate almost entirely the possibility of data snooping. Data snooping refers to the practice, consciously or unconsciously, of cherry picking only the most successful technical trade rules over some period, and not reporting those found to be
unprofitable. It also includes testing trade rules over many periods and many indices, but claiming only to use the period and index in which the rules performed the best. Researchers who have asked technical analysis practitioners about their best rules in the past fall into this trap because they will tend to report rules that worked best in the previous period when it is unlikely they would be aware of such rules before the beginning of that period. Because the AK algorithm systematically trains only in previous period information and only uses this information to trade in the test period, it avoids this pitfall by design.

One issue with the AK genetic algorithm process is that it is not deterministic and, therefore, does not inevitably converge to a single steady state result no matter the input. It does, however, tend to converge to some degree, as AK demonstrate visually with graphs showing each rule generation’s return and the tendency to converge slightly above buy and hold returns. In their paper, AK will acknowledge that in some kind of problem sets this is not a major issue, as the genetic algorithm will tend to asymptote. They state in other types of problems, when the genetic algorithm gets stuck, an additional “hill-climbing” method can be applied given the mathematical conditions for such a process to work hold. At the time AK were writing, computing power was a major impediment to their project, and they include an experimental section in which they combine the computing power of several PCs and run a genetic algorithm with a larger seed population to start over the hundred or so generations their main results use. This process takes weeks to run. Unlike the problems they cite in their defense of the genetic algorithm, the technical analysis problem does not seem to asymptote, at least to the degree the other cited projects do and at the seed population levels and generations used, as they find unambiguously higher returns in this higher seed population case.

In this paper, I take the AK framework and like several others, I modify it. However, my purpose is different from anyone before me (as far as I can tell from published papers). In the AK framework, AK uses several types of technical analysis trade rules, and builds seed trade rules from them. The most successful trade rules by some measure of success are then “mated” with some small chance of mutation, and over some number of generations the returns on these trade rules tend to converge slightly over buy and hold. HK, on the other hand, have learning strategies that limit trade rule by momentum measure and comparison value, and switch between predetermined trade rules, subjecting their analysis to the data-snooping problem. In this paper, I will limit the kinds of technical analysis trade rule tested to one very specific type, a moving average rule, according to several papers above the most successful trade rule in practice. Unlike HK, though, comparison values will be generated from moving average values that exist within the training period rather than surveying the most historically successful. In so doing, I hope to maintain the advantages of the HK and AK approach while minimizing their respective weaknesses. The Moving Average\(^1\) rule was posited in several papers before my time

---

\(^1\) While not included here due to space constraints, I have looked at Exponential Moving Average as well, which treats more recent prices as more important than less recent prices. The results are quite similar in the exact treatment of the training result, but differ in the Mean and EMA-3 treatments. MA was chosen here specifically because its results were most consistent across all treatments, implying, but not ensuring, that it captures something more fundamental to the nature of prices and technical analysis. BETC results of the EMA parameter are available upon request.
period begins (but also after) so it is reasonable to believe this assumption does not create a data snooping problem.

To start, the algorithm, in the training period, given some n-period MA trade rule, the percent difference between the current period price, and the mean of the last n periods, including the current period price, is calculated. This process creates a list of a corresponding percentage difference over the n-period MA trade rules for each of the past prices over some frequency in some time period, with the exception of the first n-1 prices. These first n-1 prices for any given n-period MA trade rules are not tradeable. Each trade rule consists of a buy-in indicator value, b, and a trade condition. The b- indicator value moves from a value of negative one to one and back, allowing the rule to buy or sell respectively, and ensures that the algorithm will have to buy in initially and only sell when its most recent past transaction was a buy. The percent difference value for each corresponding price is then compared with one of the other percent differences in the following rule:

Where c is a list of 3 to 5 stock prices.

\[ m = \frac{c_{end} - \text{mean}(c)}{\text{mean}(c)} \], m is the percent difference of final value and mean of c.

For the buy rule, if the current m percent price difference is greater than the one it is compared to, and the last buy/sell decision was a sell or it hasn’t entered the market, then buy. If the last buy/sell decision was a buy, then hold. For the sell rule, if the current price difference is less than the one it is compared to, and the last buy/sell decision was a buy, then sell. If the last buy/sell decision was a sell, then continue out of the market. For each compared to percent difference, the total return over the period is calculated using that simple technical trade rule. This process is repeated over the full training period with each individual percent difference used once as the compared to price for the full period. In a typical case, the process is also completed for several values of n and in this paper those values are three to five. At the end of this training period, there will be a list of all the simple trade rules used and the corresponding returns for each of them. By design, these simple trade rules constitute the possible returns of trade rules over the training period of the limited form and type designated by the algorithm. This makes the process deterministic, unlike the genetic algorithm process detailed in the original AK paper. Crucially, and by design, these trade rules contain all possible trade rules of the specific type tested that change the trade pattern.

The AK paper allows for several different kinds of fitness tests, and in their paper, they use primarily an equation that calculates return minus a transaction cost fee per trade made. I record the n-period MA and the percent difference compared with that has the highest return, referred to as the MA parameter later on, and this is the result of the training period. As stated before, because the process is deterministic, the results will be the same no matter how many times the algorithm is run. If more than one trade rule results in the same highest overall return, then this set of trade rules is the training result. There is no arbitrary limit, although practically speaking, in my results, it rarely goes above 15 or so trade rules.
In the trading period, there are three processes by which this training period result are used. The first is identical to AK. Simply, the n-period and percent difference result that performed the best in the trading period is run over the trading period, and the results recorded. If more than one n are used, then all n that appear in the training rule are used. In the next, the best trade rule MA comparison values are run over all n-periods considered, and the returns are averaged. This implicitly carries through results of previous papers that show trading several rules at once result in a positive diversification effect, like investing in several assets at once. Finally, the top percent differences are run only in a 3-period MA, and again if there are multiple results, they are averaged. This specific treatment comes from the original AK paper that show that 3-period MA is the best of all MA in returns. Note that unlike some other algorithms, in which the decision is made between the risk-free asset and the asset, here, the decision will be made between holding the asset and zero-return, or practically the same, a zero-return risk-free asset. Additionally, and unlike previous AK papers, this paper tests the differences in varying the length of the training and trading period over the same asset. The training periods are between one to ten given periods of time in length with trading periods of the same time afterwards. For example, given daily frequency stock price, the training periods are one to ten months with corresponding trading periods of the same length. When the stock prices are a 5-minute frequency, the training periods are one to ten days long. Like AK and others, the algorithm repeats this training and trading process, using each period as a training period for the proceeding trading period, such that the only periods not used as both a training and trading period are the first, which is used only for training, and the last, which is used only for trading. To be clear, each training period starts from scratch. No previous period information is carried over. When calculating transaction costs among several trade strategies, as happens quite commonly for the second and third results treatments above, an average is taken of the number of transactions of each strategy, a process which preserves the equal weight of each assumed by the returns averaging.

Given the tendencies of previous paper results, that the least mature markets tend to have the most profitability of technical analysis, any paper that hopes to look at the universality of technical analysis and its implications for the Efficient Market Hypothesis ought to look at one of the most mature markets. For this purpose, I focus on the Dow Jones Industrial Average. To buy the Dow Jones Industrial Average all at once in a practical sense would likely result in thirty times the transaction costs as most brokers charge per equity sold. Instead, I look at the ETF DIA as it is the closest following ETF to the Dow Jones Industrial Average available, and it is available for purchase at the same transaction cost as equities in most places. Most daily data I downloaded from the Center for Research in Security Prices or CRSP, the standard academic data source for equity pricing data. My university’s subscription did not have access to the higher frequency data, so I downloaded the 5-minute frequency data from Kibot.

Like the Dempster and Jones (2000) paper did for the FX market, but unlike the other papers cited above, I will look at the same technical analysis algorithm over two different time frequencies, daily price frequency and 5-minute price frequency. The training and trading time periods are monthly and daily respectively. An advantage of using an index composed of relatively few component equities (around 30 at any given time) is that I can realistically also test each individual equity in the same two price frequency and over the same training and trading periods. In this way, I can make some comparisons over
the time periods and equities, and over the index as a whole. Many of the papers above, including AK, experiment with including transaction costs in their algorithm/rule determining process, and so I compare adding transaction costs to both the daily and 5m time frame training of the algorithm and show how these affect the BETCs that result from it.

Finally, I provide robustness tests for these results. Unlike parameterized models, and even some non-parametric models, there are very few known issues like autocorrelation, heteroskedasticity et cetera in genetic algorithms. The main issue in tests of technical analysis, data-snooping, is eliminated by the design of genetic algorithms (and I argue, also, my learning strategy in the genetic algorithm framework). Still, there are two assumptions implicit in the experiment design that can be robustness tested to some degree. The first is that all purchases the algorithm desires to purchase or sell will certainly go through. For example, if there is a substantial delay between the computer receiving the price, determining whether it will purchase or sell and actually sending the purchase order, then it is possible the purchase doesn't go through. This “significant delay” can be on the order of hundreds of milliseconds in the real world. This is accomplished by sending a conditional order, in other words, purchase or sell only if the price is exactly (or less than or greater than) this amount. If some don’t go through, then perhaps the expected outcome in terms of Break-Even Transaction Cost (BETC) will vary. To this end, I run 20 simulations of each training and trading sequence 1 to 10 n periods of the top performing parameterization (MA with transaction costs) when a trade (buy or sell) randomly goes through only 75% of the time. I report then, the top 5%, bottom 5% and mean BETCs over all time-periods as well as compare them to the BETCs that were acquired over the same period when 100% of trades were assumed to go through.

Finally, the algorithm makes the implicit assumption that prices reported by the index at a specific time are the prices at which a trader could purchase or sell the given equity at that exact moment. Technically, this may not be the case, as indices actually report the prices for the most recent transaction, rather than the most recent offers to buy or sell some set number of that stock. By allowing the algorithm to see completed transaction prices, but requiring it to purchase or sell only the prices currently bid and asked at, we require the algorithm to work in more realistic conditions. There are limits to the second-by-second data I have, discussed in more detail below, but it is still more realistic than the original algorithmic process implicitly assumed. I should note that finding and purchasing data with this level of granularity is not trivial. Special thanks goes to my dissertation advisor Dr. Erick Rengifo for providing the data I used.
Results:

I. Comparing the Break Even Transaction Costs (BETCs) of Moving Average (MA) at the 5-minute time period vs. daily

a. BETCs by Index, Component and Time-Period:

Figure 2a – 2d – BETC for the ETF DIA

Figure 1a

Figure 1b

Figure 1a – 1b shows the Break Even Transaction Cost (BETC) ETF DIA, which is the most closely tracking ETF to the DJIA available, over the 5-minute time frequency in Figure 1a and over the daily time frequency in Figure 1b. The color of each bar represents the three training result treatments in this paper, where blue, green and yellow represent, the original AK treatment, all top strategies over all possible time periods, and only the top return strategy on only the 3-period rule, respectively. Returns are remarkably consistent across training types and training periods in the 5-minute frequency, while significantly negative in the daily, except for four exceptions in each, in each case remaining positive for all three of the 7-day length training/trading period.
Figure 2a – 2b shows the mean Break Even Transaction Cost (BETC) over all 40 stocks for which 5-minute frequency data was available in the component stocks of the DJIA and over 70 stocks for which daily data was available, over the 5-minute time frequency in Figure 2a, and over the daily time frequency in Figure 2b. The color of each bar represents the three training result treatments in this paper, where blue, green and yellow represent the original AK treatment, all top strategies over all possible time periods, and only the top return strategy on only the 3-period rule, respectively. Returns are remarkably consistent across training types and training periods in the 5-minute frequency from three to ten training/trading periods, while significantly negative in the daily in all cases.
b. Ranking Strategies by Mean percentage gain over time deciles of component stocks strategies.

Figure 3a

Figure 3a ranks the 30 strategies used here by percentage gain in the 5-minute time frequency, taken as an average of component stock results. This is to measure the persistence of the percentage gains over time (here separated into time deciles). A perfectly persistent graph would have all strategies with a horizontal line at their consistent rank. While these lines are clearly not horizontal, the MA parameter displays striking rank persistence within each n training/trading day strategy group. The top three strategies are the n = 10 cluster of strategies.

c. Ranking BETCs over time, by rank and value:
In Figure 4a and 4b, I measure the BETC of the DIA ETF 5m time period by rank and value over ten deciles of time. In Figure 4a, unlike the mean ranks of the DJIA component stocks, there is no consistency at all in the ranks of the strategies, however, their respective values in Figure 4b are extremely close to each other. Figure 4b, has a high peak at decile 1 of 4 cents, a precipitous drop, and BETCs between .5 and slightly above 1 cent the rest of the ways. This result suggests the theory that markets become more efficient over time.

In Figure 4d, I graph the BETC over time for the MA parameter. In Figures 4c, I use each of the 30 strategies’ median rank and graph it over the 10 time deciles. Besides the top three strategies by rank, (n=10 training/trading days), the rankings are quite scrambled and very erratic. However, Figure 4d shows the BETC values over time deciles and these demonstrate that the BETCs are quite consistent in both cases, with three distinct peaks over time, each of which peak around 9, 7 and 6 tenths of a cent respectively, in both cases.

II. Comparing ½ cent versus zero transaction costs in the algorithm training:

Below, I compare the BETCs with a ½ cent vs. the zero cent transaction cost results from above. The three training result treatments are averaged in every case, to simplify interpretation. In most cases, the BETCs of the three treatments are quite similar to each other. The transaction cost in training results are provided in full in the appendix.
a. Comparing BETCs in the ETF DIA and Component Stocks:

In Figure 5a and 5b, I compare the BETCs of the zero transaction cost in training, versus ½ cent per transaction cost in training, where the blue bars represent zero transaction costs and the yellow bar represents ½ cent transaction costs in the ETF DIA. In Figure 5a, the BETCs are reported for the 5-minute price frequency. In every case, adding the transaction cost increases BETCs. In some cases, by as much as .2 or .3 cents. In Figure 5b, I report the BETCs for the daily price frequency. The results here are much less conclusive. In four of ten cases, the BETCs increase. In two of ten cases, the BETCs are more or less the same. In four of ten cases the BETCs decrease. One interesting thing to note is that the BETC goes from negative to positive in the case of n=10 Trading/Training months.
In Figure 5c and 5d, I compare the BETCs of the zero-transaction cost in training, versus \( \frac{1}{2} \) cent per transaction cost in training, where the blue bars represent zero transaction costs and the yellow bar represents \( \frac{1}{2} \) cent transaction costs in the component stocks of the DJIA. In Figure 5c, the BETCs are reported for the 5-minute price frequency. In every case (except \( n = 1 \)), adding the transaction cost increases BETCs. In some cases, by as much as 1 or more cents. In Figure 5d, I report the BETCs for the daily price frequency. The results here are much less conclusive. In almost every case there is very little movement, with slight improvement in the \( n = 2 \) case.

b. Measuring Persistence of BETCs in the 5-minute Results:

From the results above, it is clear that the 5-minute results are substantially more consistent, showing clear improvement in almost every case from no-transaction costs in training to transaction costs, and consistently positive BETCs across the board. Here I determine whether these results are persistent across all ten time-deciles of the five-minute frequency, MA parameter, with transaction costs included in training for both component stocks and the ETF DIA.
In Figures 6a, 6b and 6c, I measure the persistence of BETCs over time decile in the 5-minute MA parameter ETF DIA. Figure 6c measures the max and minimum BETC each decile. These effectively represent an interval within which any agent who does not run all the parameterizations (trying perhaps to run only those most profitable) at once must occur. The blue line represents the maximum strategy, and it is above the .5 cent BETC in every time decile. The orange line, representing the minimum strategy, is below the .5 cent BETC line three times, in the 4th, 5th, and 10th time decile. Figure 6a represents the mean of all 30 parameterizations at each time decile. Nine of ten deciles are above .5 cent BETC, with the fourth time decile exhibiting a .49 cent BETC, just below .5 cents. Figure 6b represents the median of all 30 parameterizations at each time decile. Every decile exhibits a BETC above .5 cents.

![Figure 7a](image1)
![Figure 7b](image2)
![Figure 7c](image3)
In Figures 7a, 7b and 7c, I measure the persistence of BETCs over time decile in the 5-minute MA with TC parameter of the Dow Jones Industrial Average component stocks. Figure 7c measures the max and minimum BETC each decile. These effectively represent an interval within which any agent who does not run all the parameterizations (trying perhaps to run only those most profitable) at once must occur. The blue line represents the maximum strategy, and it is above the .5 cent BETC in every time decile. The orange line, representing the minimum strategy, is below the .5 cent BETC line eight times, all but the second and fifth decile. Figure 7a represents the mean of all 30 parameterizations at each time decile. Eight of ten deciles are above .5 cent BETC, with the eighth time decile exactly equal to .5 cents BETC and the ninth time decile equal to .37 cents BETC. Figure 6b represents the median of all 30 parameterizations at each time decile. Every decile exhibits a BETC above .5 cents except the ninth, with a BETC of .38 cents.

III. Robustness Test One – Simulating a 75% Successful Purchase Rate:

The results above show several cases of eight, nine or even all ten time deciles exhibiting BETCs that violate the efficient market hypothesis. Here, I robustness test these results by randomly disallowing 25% of desired purchases (or sales) the algorithm wants to make. If disallowed, the algorithm must try to purchase (or sell) the next period under the same requirements, if the trade rule still holds true. The process is run twenty times and the maximum, minimum, median and mean BETCs are taken, to indicate the interval BETC range, and expected values in these cases. This is meant to more realistically simulate market conditions, when desirable prices may only exist for a very brief period of time.

a. Comparing BETCs by training/trading days in Sure and Simulation Purchase Cases

![Figure 8a](image-url)
In Figure 8a, I compare the original Moving Average with Transaction Costs for the component stock trade rule parameter over all \( n = 10 \) training/trading days, where the blue bars represent BETCs when the algorithm is able to complete all desired trades, while the green bar represents the mean of 20 simulations in which only 75% of desired trades go through. The difference between the sure and sim cases is relatively constant around 5/100ths of a cent. This is crucial for \( n = 7 \) to 10, where the sure case is greater than \( \frac{1}{2} \) a cent in all four cases, but with the simulation, it is only greater than \( \frac{1}{2} \) a cent once.

![Figure 8a](image.png)

In Figure 8b, I compare the original Moving Average with Transaction Costs for the ETF DIA rule parameter over all \( n = 10 \) training/trading days, where the dark blue bars represent BETCs when the algorithm is able to complete all desired trades, while the green bar represents the mean of 20 simulations in which only 75% of desired trades go through. The difference between the sure and sim cases is relatively constant around 1/10th of a cent, larger than the previous case, with sure always exhibiting higher BETCs.

![Figure 8b](image.png)
b. BETCs over Time – Sure vs. Sim Case:

In Figure 9a I report the mean of all parameterizations with sure transaction MA TC 5m for all component stocks of DJIA, represented by the red line, and the mean simulation represented by the blue line, all over ten time deciles. Qualitatively results are largely the same, with a BETC of .5 cents at decile 8 in both cases, and one BETC below .5 cents at decile 9. Quantitatively, the simulated results have a slightly higher variance, but BETCs remain largely the same. In Figure 9b, I report the median of all parameterizations with
Sure transaction MA TC 5m for all component stocks of DJIA, represented by the red line, and the median simulation represented by the blue line, all over ten time deciles. Again, qualitative results remain the same, with only one BETC below .5 cents in decile 9 in both the SIM and Sure case. In Figure 9c, I report the Max-Sure represented by the black line, the Max-Sim represented by the green line, the Min-Sure represented by the red line, and the Min-Sim represented by the blue line. Again, qualitative results remain largely the same, with all Max BETCs in the SIM and sure case above .5 cents, and only the second and fifth decile above .5 cents of the Min BETCs in the SIM and sure case above .5 cents. Quantitively, numbers are quite similar, with a slight increase in variance between max and minimum of the sure to the sim case.
In Figure 10a I report the mean of all parameterizations with sure transaction MA TC 5m for the ETF DIA, represented by the red line, and the mean simulation represented by the blue line, all over ten time deciles. Qualitatively results are largely the same, with a BETC of slightly below .5 cents only in the fourth decile. Quantitatively, the simulated results have a slightly higher variance, but BETCs remain largely the same. In Figure 10b, I report the median of all parameterizations with sure transaction MA TC 5m for all component stocks of DJIA, represented by the red line, and the median simulation represented by the blue line, which both have BETCs all over .5 cent in all ten time deciles. In Figure 9c, I report the Max-Sure represented by the black line, the Max-Sim represented by the green line, the Min-Sure represented by the red line, and the Min-Sim represented by the blue line. Again, qualitative results remain largely the same, with all Max BETCs in the SIM and sure case above .5 cents. In the minimum simulation case, there are four cases below .5 cent BETC, and three in the sure case. Still, numbers are quite close to each other in general, with differences in the fourth case less than $1/10^{th}$ of a cent between the sure and simulation case. Quantitatively, other numbers are quite similar, with a slight increase in variance between maximum and minimum of the sim and sure case respectively.

**IV. Robustness Test Two – Using Real Bid-Ask Data**

As mentioned above, the value listed as the price of any given stock on exchanges merely lists the price at which the most recent transaction of that stock took place. It does not list the price at which one could currently sell or buy the stock. Instead, there are real time bids, that is, prices at which a stock trader is willing currently to purchase a given stock for, and real time asks, prices at which a stock trader is willing currently to sell a given stock for. To realistically robustness test the differences in prices that have occurred and prices that could be attained, I acquired 5 trading days of bid-ask data for the Dow Jones Industrial Average. These were February 19, 20, 21, 22 and 25 2019. The bid-ask data listed the most recent bid and ask price on a second long time-scale for the entire trade day on each of these five days. As prices can change on a microsecond time scale, particularly in recent years, this is still an imperfect robustness test. A quick look at the data shows the dramatic variation the prices can have even on this very small-time scale, where the y-axis represents the size these micro-fluctuations can take. When half a cent is the transaction cost, even a 1 cent fluctuation can be meaningful, much more the 5-10 cent fluctuation from second to second, seen below in Figure 11a over a single day.
To get an idea of the period the algorithm is running, I collected data surrounding the five-day period over which the analysis took place. I chose to start January 2\textsuperscript{nd}, 2019, the first day of market activity in January 2019, and ended the day I collected this data, March 6\textsuperscript{th}, 2019.

\textit{a. BETCs over the analysis period:}

In Figure 11a and Figure 11b I show the BETCs over the longer time period, which are significantly negative, even in the MATC trade parameter, a result suggesting that markets are becoming ever more efficient. Figure 11b features the BETCs when all 39 stocks are included, while Figure 11c shows the data when the 35 of 39 stocks for which at least one successful purchase and sale went through over the five-day period. BETCs are largely the same, but slightly more negative in the first case.
b. Transaction Success during Bid-Ask test:

During this robustness test, and days for which bid-ask data was available, each trade the algorithm desired was then run through an additional program that simulated a standard bidding process. This program would look for an ask (the minimum price a seller is willing to sell at) that was equal or lesser than the price the algorithm desired to buy at, and a bid (the maximum price a buyer is willing to pay) that was equal or greater than the price the algorithm desired to sell at. If such a desired price was found, then the transaction went through, at the dollar amount of the bid or ask. If it was not found within the 300 second-by-second prices listed between 5-minute prices the algorithm was run over, then the transaction was denied, and the algorithm was not allowed to complete that transaction. Taking the thirty-nine of forty DJIA stocks for which bid-ask data was available, there is a successful sell rate of 23.29%, a successful purchase rate of 96.16%, and an overall successful transaction rate of 36.76% for the five-days studied. When omitting companies for which there were no successful bids (sales at a desired price) at all over the five-day period, the successful sell percentage is 39.76%, the successful purchase rate is still 96.16% and the total transactions through is 55.5%.
c. Comparing Bid-Ask BETCs with 100% transaction success:

Next, I show the returns over this five day period using the Bid-Ask purchase program versus the old assumption that 100% of trades go through at the price listed on the index (including all 39 stocks for which data was available). As the very low successful transaction rate implies, the BETCs are substantially different, with no positive trade rules found in the standard Bid-Ask return. Below, in Figure 13a, the 5-day Regular return, I find positive BETCs reaching as high as 10 cents, but also as low as -25 cents. In Figure 13b, the 5-day Bid-Ask return, there are BETCs as high as 11.9 cents, and as low as -15 cents. Taking the mean in both cases I can compare the result. The Mean BETC for the regular purchase process over this period is -0.0655 dollars while the mean BETC for the Bid-Ask purchase
process is -0.0249 dollars, beating the standard over this period. This implies that the Bid-Ask process BETC improvement effect from obtaining bid and ask prices better than the algorithm desires is stronger than the BETC loss effect that would come from not always successfully completing transactions.

Conclusion:

Past papers that looked at technical analysis had several problems. The core problem of so-called “data snooping,” involved analysts intentionally or unintentionally finding rules that were profitable in the past and implying agents in that period could know them before such a time. This typically involved either asking technical analysts about past profitable rules or simply running several set rules over the previous period until one was found to be suitably profitable. This problem was solved with the genetic algorithm, which only determines trade rules using past information, and it is also handled by the unique learning algorithm in a genetic algorithm framework I demonstrate here.

A crucial discussion surrounds the Break-Even Transaction Costs (BETCs). Nearly all past papers say the average Break-Even Transaction Cost is between 5 and 50 basis points, that is between .05% and .50% of the price of the equity in question. While this may have been a very realistic way to measure transaction costs in the past, in the modern era, transactions often happen in the blink of an eye, with no substantial difference in the technological marginal costs between an equity that costs $1 per share and another that costs $300 a share (both values are contained in my sample). A transaction cost of any set basis point implies then that the transaction cost of the $300 share is 300 times that of the $1 share, a simply untenable assertion in this day and age of computers. Rather, over the past twenty years, the prices investors, particularly those interested in high volume, pay per transaction are often fixed. A standard rate for US equities is half a cent per share per transaction, while volume discounts can be significantly lower (Reinkensmeyer).
Interactive Brokers, the apparent standard setter in low-cost transaction prices, offers volume discounts as low as .0005 dollars (or .05 cents) per share per transaction (Commissions). Still, as most investors are not likely to be able to take advantage of these prices, as the volumes required are quite high, and in order to be conservative, this paper will consider only their standard price of half-a-cent per share per transaction. Additional considerations involving transaction costs are that some high frequency trading brokers have no transaction fees applied to ETFs. Interactive Brokers, for example, has 48 commission-free ETFs currently available, a relatively low number in that space, and particularly important due to the DIA ETF’s consistent positive BETCs. Additionally, some indices give “maker-taker” high frequency trade rebates that increase liquidity, resulting in an effectively negative transaction cost.

The highest bar for technical analysis is a process which beats buy and hold with realistic (and ideally higher than break-even) transaction costs, and that is persistent enough such that an investor could realistically choose it before the period which it profits to this degree. Several papers before mine, including all of the genetic algorithms/programs cited here, have demonstrated technical analysis trading patterns which beat buy and hold with no transaction costs. In this paper, all the 5-minute frequency trading rules beat buy and hold, both in the ETF DIA and the mean of the Dow Jones Industrial Average (DJIA) component companies. Very few of the daily-frequency trading rules beat buy and hold, and those that did either beat it only very slightly or are not persistent enough to be exploitable. This explicit comparison of the same technical analysis algorithm over the same assets in different time periods is, as far as I know, unique in the equity market, and has only been done before in the FOREX market, by Dempster and Jones. The results of this explicit comparison are quite clear. In the daily frequency, technical analysis of the moving average parameter tested here is of no practical economic value. These conclusions are similar to other papers, ie Bajgrowicz and Scaillet (2012) and Hsu and Kuan (2005), that find no technical analysis trade patterns that beat Buy and Hold in the Dow Jones Industrial Average. Bajgrowicz and Scaillet (2012) uses the lower comparison bar of the Federal Funds Rate as a riskless interest rate and come to the same conclusion. Another unique contribution is the comparison of the training/trading periods. Here there is a general trend that longer training and trading periods are consistently the best returning and highest BETCs for the MA parameter, suggesting a clear future research value in testing eleven day/month or higher length training/trading periods.2

Next, there is the issue of transaction costs included in training periods. This has been done several times before in the literature, including in the seminal AK paper on the topic. Here, I compare including a zero-cost transaction included in training versus a half-cent transaction cost included in training, and compare results. For both the MA parameter over the ETF DIA and over its component stocks, BETCs increase, between .1 cents to .3 cents typically, and by as much as 1 cent in some cases.

An additional issue is the persistence of the profitability of a one-time profitable trade rule. For example, Ready (2002) demonstrates quite convincingly that the BLL (1992) results showing a period of profitable technical analysis in the Dow Jones Industrial Average from 1963 to 1986 was not persistent, and in fact, both before and after this period, the BLL rules result in significant losses. For any technical analyst, the rules studied

---

2 Initial tests suggest that this improvement from longer training periods asymptotes to zero by n = 20.
must be persistent and knowable, neither of which apply to the BLL results, Ready contends. Here, I take the mean and median rule parameterizations for the 5-minute MA parameter with transaction costs included\(^3\), over the ETF DIA and the DJIA component stocks. In the case of the mean rule over the ETF DIA, nine of ten deciles exhibit BETCs above or equal to the realistic BETC of .5 cents, with the tenth exhibiting BETC of .49 cents. In the case of the median rule over the ETF DIA, all ten deciles exhibit BETCs above .5 cents. In the case of both the mean and median rule over the mean of the DJIA component stocks, nine of ten deciles exhibit BETCs above or equal to .5 cents, with the remaining tenth in each case above .3 cents. Even given the substantially more stringent requirements for an efficient market hypothesis breakdown that are used here, there is evidence that the EMH fails over the 5-minute time period, in at least 16 of the last 20 years, over what most economists would consider the most mature stock market index in the world.

There are several further considerations, but none has the potential to invalidate these results as slippage does. Slippage refers to an algorithm or trading system, typically of medium or high frequency, missing a desired trade price because it doesn’t exist for a long enough time to capture it. As equity prices are now determined on at least a microsecond time-frame and prices may only exist for such a period, this is not a trivial problem. Authors like Dempster and Jones in the FOREX market have considered slippage, but this is the first time it has been considered in equity markets, likely because technical analysis has shown little promise above BETCs in the literature, particularly in mature markets. I robustness test for slippage in two main ways, via simulation and Bid-Ask.

In the first, I take the 5-minute Moving Average trade rules with transaction costs included in the training for the ETF DIA and component stocks of the DJIA and randomly allow only 75% of desired trades to go through during trading periods. This is done over 20 simulations, and I take a mean (here, also the weighted mean, and thus expected value) of the BETCs. I find the simulation overall results in a 5/100 cent BETC reduction in the component stocks of the Dow Jones and a 1/10 cent BETC reduction in the case of the ETF DIA. Taking persistence into account, in the persistence measure of the ETF DIA, and component stocks, variance of BETCs increases slightly in the simulation, but qualitatively results remain the same, with only one decile having a BETC below .5 cents in the mean case, and none in the median case. In the case of the component stocks, the mean now has two BETCs below .5 cents (one being the exactly .5 cent BETC in the sure case), while the median still has only one BETC below .5 cents. So, while simulation increases variance in the results, qualitatively the BETCs are still largely above realistic transaction costs in the substantial majority of time deciles.

Stock market indices list prices, such as those used in this paper, but they only list the prices of completed transactions, not prices a trader could necessarily purchase or sell at right now. In the second robustness test, I take 5-days of bid data, which lists prices a bidder is willing to pay for a stock, and ask data, which lists prices an asker is willing to sell for, and require the algorithm to go through this bid and asking process when it wants to purchase or sell. When taking all thirty-nine stocks and all transactions into account, the algorithm successfully purchases or sells when desired only 37% of the time over the 5-day

\(^3\) I include the results when .5 cent transaction costs are included in the training. The results for a 1 cent training transaction cost (assuming .5 cent actual transaction costs still) are quite similar quantitatively and are available upon request.
period, a substantial difference from the 100% assumed in the main results. However, BETCs actually increased from the main results, as the purchase program allowed a trader to buy at a lower cost than desired and sell at a higher cost than desired, if such a price was listed in the second-by-second bid-ask data. Still, in both cases BETCs were substantially negative and they improved only from -6.55 cents to -2.49 cents. These results are by no means comprehensive, particularly due to lack of data, but they do suggest that including Bid-Ask robustness does not decrease BETCs, and may actually increase them.

There are several further considerations here. Dividends were not included in this analysis, a common simplification that all but one of the papers in the literature from above make, and one that would increase Buy and Hold returns relative to any trade rule returns. Still, theoretically, if the dividend pay-out day is known to an agent, he could simply stay buy and hold on the day the dividend is issued and trade the rest of time, which would likely change net results very little. Additionally, I assumed a trader, when out of the market, received zero-return, rather than a riskless positive interest rate, an inclusion that would increase BETCs. In order to keep prices consistently representing the same proportion of the company, stock split adjusted prices were used, however, this means that practically speaking, more than one stock purchase (or less in some cases) would need to be made, and the per stock transaction prices would change multiplicatively at this rate, in some cases eliminating BETC above the realistic TC, while in others it would increase it. The implicit assumption made here, that the same BETC would occur scaling down prices to one stock before the adjustment is a large assumption to make, and one that deserves further research. Other papers solve this problem by making the unrealistic assumption that transaction costs are best measured with basis points. There are strong assumptions made in each transaction cost measure. Another issue is the change in quantity of supplied and demanded orders of these stocks would make, potentially changing stock prices in the future from the fixed prices the analysis implicitly assumes. The implicit assumption is that the orders the algorithm make are small enough that they do not affect price. This assumption increases in strength the more stocks that are ordered, but this also decreases transaction costs with volume pricing. Still it is an issue quite challenging to test without a real-world experiment. Finally, in order to simplify the analysis, buy and hold over any period is recorded as the difference between the last and first price over any time period. In a ten-day training/trading period, for example, the first ten days is always training, so the algorithm is not allowed to invest or return anything at all, while these returns are included in Buy and Hold. Thus, given the average daily slight stock price increase, there may be systematically lower than realistic BETC returns measured in these cases.

Taking all these results together, there were nine deciles out of ten of EMH violation based in the mean and all ten in the median ETF DIA trade rules for the MA parameter with transaction costs included, over the five-minute stock price frequency. There were eight deciles out of ten violating EMH on the mean component DJIA stock MA, and nine on the median component DJIA stock MA parameter with transaction costs included, over the five-minute stock price frequency. All results are from 1998 to 2018, with each decile representing roughly 2 years of data. These results are qualitatively robust to slippage, in both a simulation of 25% unsuccessful trades and a bid-ask process. It is important to note that while the Bid-Ask process increased BETCs, it also substantially decreased the number of successful transaction costs, which would decrease expected total profit over Buy and Hold also. It is important to note that using some other papers’ standard for EMH violation,
positive BETCs over realistic transaction costs for some period, the EMH is violated here for all thirty parameterizations in the ETF DIA MATC 5m results, in both the original algorithm run and the simulation. It is also violated in \( n = 7 \) to 10 training/trading days in the component stocks for MATC over 5m, and \( n = 10 \) in the case of the simulation. Adding my additional requirement of persistent profit, using the median rule in the ETF DIA 5m the EMH is violated in all ten-time deciles, while other summary statistics for component and ETF DIA for indicate violation in eight to nine time deciles. These results are clearly on the cusp of violating the EMH, while the ETF DIA 5m median is a marginal violation. Still, the mean rule is a better summary statistic here, as a trader would most likely take all parameterizations and equally weight them, as determining the median trade rule would require future knowledge. Thus, while these results are very close to the margins of EMH violation, and may represent the best a technical analyst can do in the DJIA, they ultimately support a weak Efficient Market Hypothesis.

Appendix I:

a. BETCs with Transaction Costs Included in Training

In Figure A1, I find the BETCs for the MA parameter in which a \( \frac{1}{2} \) cent transaction cost is included as a penalty per trade in the training of the algorithm, when applied to the ETF DIA daily 5m prices. In these cases, BETCs are consistently above 1 cent, even in the \( n = 1 \) training/trading parameterization, which is often the worst performing in previous parameterizations. In the case of \( n = 10 \), the exact parameterization (designated by the blue-green bar in the middle) reaches almost a 1.5 cents BETC. In Figure A2, I report the MA parameter with transaction costs in the daily time period. There are six of 30 parameterizations with a positive BETC, including all three treatments of \( n = 7 \), peaking at around 3 cents BETC for the MA-3 treatment.
In Figure A3, I find the BETCs for the MA parameter, when a \( \frac{1}{2} \) cent transaction cost is included in the algorithm training periods. I find almost all BETCs above \( n=7 \) are above the \( \frac{1}{2} \) cent transaction cost for the full period, with the exact treatment of the \( n=10 \) training/trading days parameterization achieving almost a .6 BETC. In Figure A4, I show the BETCs for the daily mean treatment, and there are no positive values here with the daily parameter.

b. BETCs over Time Deciles

In Figure A5, I record the BETC by decile for the MA parameter with transaction costs. I find three peak BETCs of around .9, 1 and .8 cents in the second, fifth and last decile. There are also two troughs, in which BETCs for the lowest parameter go as low as .2 cents. Like before, none of them become negative, but also even the lowest time deciles are typically slightly above the .5 cent break even transaction cost, particularly when the mean
BETC is taken (see above). Figure A6 reports the ranks of these parameterizations by BETC, and shows very little consistency as usual in the middle, but slightly more at the highest rank average (lowest BETCs) and a fair amount, with the same top three almost every decile, with the lowest rank average (highest BETCs).

In Figure A7, I record the BETCs by decile of the MA parameter with transaction costs on the 5-minute price frequency of the ETF DIA. Like before, there are extremely high BETCs in the first-time decile, as high as 4 cents, before decreasing and settling typically between .5 cents and 1 cent, with a dip to .4 cents in the fourth and fifth decile.

Appendix II – Expanded Simulation Results:
In Figure A8 I report the mean BETCs with the MA parameter of component stocks, using transaction costs in training (MATC) over the 5-minute time frame, taking the mean BETC of all 20 simulations (Sim). The purpose here it to demonstrate whether the mean BETC in this simulation, where trades go through 75% of the time, is similar to the BETCs that are found when purchases go through 100% of the time. I find similar, but slightly lower BETCs, and with BETCs of n=7 to 10 above .45 cents, and n = 10 exact treatment above .5 cents.

In Figure A9, I report the mean BETCs with the MATC parameter, over the 5-minute Sim, and find results that are very similar to the MATC in which trades went through for sure, but slightly lower on average. Peaks are at 1, 1.1 and .9 cents, in the 2nd, 5th and last time decile respectively, with a trough at the ninth time decile with a BETC slightly below .2 cents. In Figure A10, I report the mean ranks of these MATC-Sim BETCs, and find they are inconsistent, except for the highest ranks (ie ranks worst BETCs) and some consistance in the lower ranks as well.
In Figure A11, A12 and A13, I report the BETCs of the MATC parameter, of each of the three training treatment cases respectively. In the first case A11, I report the Mean, then A12, the Exact and finally A13, the MA-3 treatment. For each training/trading case n value, there are four values, representing the minimum simulation BETC, the “sure” BETC, the case where all desired trades go through at the exact price desired, the mean BETC, that is the mean of all the 20 simulation BETCs and the max simulation BETC. While the minimum and maximum are interesting in their providing a reasonable floor and ceiling range, in some sense corresponding to the middle 90% (maximum and minimum are each the bottom and top of twenty, or the top and bottom 5%), the comparison of the mean and sure BETCs is the key comparison. That is, if only 75% of trades go through, does that dramatically alter BETCs. While the “Sure” BETC is higher in all cases, particularly in the mean treatment, it is not substantially so, and only differs by 1 to 2 hundredths of a cent less, particularly in the exact and MA-3 treatments. Note also that the minimum and maximum returning trade simulation is chosen by the maximum raw return of the treatment, rather than by minimum or maximum BETC.
In Figure A14, I report the BETCs of the Mean MATC-Sim parameter on the DIA ETF in the 5m price frequency case. I find again that overall BETCs decrease by about one hundredth of a cent, peaking at 1.5 cents in the sure case, and 1.4 cents in the mean case.

In Figure A15, I show the BETC MATC-Sim of the 5m time period over ten time deciles. I find it qualitatively has a very similar shape with the sure case, with slightly lower BETCs in every time decile.
In Figure A16, A17 and A18, I show the BETC MATC-Sim ETF DIA of the 5m time period split into the Mean, Exact and MA-3 training treatments respectively, ranked by the raw return. Here, again, like before, the sure treatments are slightly higher than the mean treatments, by one to two hundredths of a cent, with the Exact treatment having the highest differences. Interestingly also, the Mean Simulations are the only one that don’t have any negative simulation returns even in the minimum case, while the Exact and MA-3 cases do.
Works Cited:


Cruz, Gustvao Galindo. “Effectiveness of Active Equity Management in Institutional Universes.” *Fordham University*, 2018.


