2016

Challenging the Efficient Market Hypothesis with Dynamically Trained Artificial Neural Networks

Kevin M. Harper

University of North Florida

CHALLENGING THE EFFICIENT MARKET HYPOTHESIS WITH DYNAMICALLY TRAINED ARTIFICIAL NEURAL NETWORKS

by

Kevin Harper

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

December, 2016

The thesis "Challenging the Efficient Market Hypothesis with Dynamically Trained Artificial Neural Networks" submitted by Kevin Harper in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date

Dr. Sherif A. Elfayoumy
Thesis Advisor and Committee Chairperson

Dr. Ching-Hua Chuan
Committee Member

Dr. Pieter de Jong
Committee Member

Accepted for the School of Computing:

Dr. Sherif A. Elfayoumy
Director of the School

Accepted for the College of Computing, Engineering, and Construction:

Dr. Mark A. Tumeo
Dean of the College

Accepted for the University:

Dr. John Kantner
Dean of the Graduate School

CONTENTS

FIGURES

TABLES

ABSTRACT

A review of the literature applying Multilayer Perceptron (MLP) based Artificial Neural Networks (ANNs) to market forecasting leads to three observations: 1) It is clear that simple ANNs, like other nonlinear machine learning techniques, are capable of approximating general market trends 2) It is not clear to what extent such forecasted trends are reliably exploitable in terms of profits obtained via trading activity 3) Most research with ANNs reporting profitable trading activity relies on ANN models trained over one fixed interval which is then tested on a separate out-of-sample fixed interval, and it is not clear to what extent these results may generalize to other out-of-sample periods. Very little research has tested the profitability of ANN models over multiple out-of-sample periods, and the author knows of no pure ANN (non-hybrid) systems that do so while being dynamically retrained on new data. This thesis tests the capacity of MLP type ANNs to reliably generate profitable trading signals over rolling training and testing periods. Traditional error statistics serve as descriptive rather than performance measures in this research, as they are of limited use for assessing a system's ability to consistently produce above-market returns. Performance is measured for the ANN system by the average returns accumulated over multiple runs over multiple periods, and these averages are compared with the traditional buy-and-hold returns for the same periods.

In some cases, our models were able to produce above-market returns over many years. These returns, however, proved to be highly sensitive to variability in the training, validation and testing datasets as well as to the market dynamics at play during initial deployment. We argue that credible challenges to the Efficient Market Hypothesis (EMH) by machine learning techniques must demonstrate that returns produced by their models are not similarly susceptible to such variability.

Chapter 1

INTRODUCTION

1.1 Equity Markets and the Pursuit of Returns

The US stock market is by far the largest equity market in the world. By some accounts, US equities represent as much as 54% of global market capitalization [Goldstein15] The market capitalization of stocks listed on the New York Stock Exchange (NYSE) reached more than $20 trillion in 2015 alone, and NYSE total trading volume in January of 2016 topped 42 billion shares [NYSE16] Market participants- be they fund managers, institutional investors, hedge funds of various sizes, or retail investors- operate in the equities markets (among other markets) for the purposes of earning returns on their money. The higher the return the better, and market participants have always and continue to look for advantages that will help them maximize this return. The term edge refers a trading advantage allowing a participant to outperform other investors, in general, and to outperform the market rate of return in particular. Yet, despite a large body of theory and research devoted to the study of financial markets, sustainable trading advantages have proven elusive for most market participants [Brown95].

The limitations of financial forecasting models have been made manifest not only by spectacular collapses of firms such as Long-Term Capital Management in the late 1990s-

a hedge fund co-founded by two Nobel Laureates in Economics, which flourished briefly by exploiting some of the newest and most esoteric financial theories of the time, but also by the observation made above: fund managers rarely outperform benchmark rates of return reliably [Lowenstein00].

The advent of machine learning techniques provided obvious candidates to aid the development of financial forecasts. In particular, Artificial Neural Networks (ANNs), with their ability to handle nonlinear processes without the need to specify model parameters, showed promise in improving these models. However, the extent to which such tools can provide a sustainable trading advantage over other market participants to extract excess returns is not clear. One reason for this lack of clarity, we argue, is that much of the research done with ANN forecasting models employs methodology unsuited to this pursuit.

A bedrock theory of finance, The Efficient Market Hypothesis (EMH), claims there are no persistent market advantages to be had. A more detailed explanation of this theory is presented in chapter 2, but the upshot of EMH is that the nature of markets is such that we should not expect any of these techniques to provide us with a reliable edge. This refutation of the efficacy of market strategies is not limited to mathematical models. It extends to all manner of financial planning with designs on beating the market (or market index) rate of return. Yet, a massive industry exists to do just this. Mutual fund managers, boutique hedge funds and financial advisors of all stripes market themselves as gatekeepers to esoteric financial wisdom. Implicit in the very idea of such wisdom is

a rejection of EMH, for if market returns cannot be reliably exceeded there is no need for financial advisors or active fund managers. Basket funds mirroring one or another of the major indices are the only logical investment choices in such a scenario.

While informational asymmetries and sporadic analytical advantages may provide some participants with a temporary edge, even the weakest form of EMH holds that such an edge is unsustainable and thus unreliable for purposes of modelling future returns. The sheer number of participant's means there will always be a few managers with multi-year track records of beating the market, but these high flyers always seem to eventually get pulled back into more earthly orbits [Goetzmann94]. An oft cited (and somewhat derisive) analogy, offered in its original form by [Malkiel99], where a group of blindfolded monkeys throw darts at a board populated with the names of listed securities helps explain this phenomenon- at least in part. If each monkey throws, say, 20 darts and so selects 20 securities, after a year half of these monkeys will have outperformed the other half as "stock pickers". At the next annual dart throwing/stock picking monkey retreat, half of those monkeys who outperformed their peers the first year will do so again the next year. After the third year, there will be several monkeys with a 3-year track record of outperforming their peers. Inevitably, this fact will be featured prominently on their firms' prospectuses to attract new monkey investors. But by years four and five, most of these hot streaks will have stalled.

In the real world, it may be that some stellar track records have to do with the skill of real, human fund managers and/or with the suitability of their investment approach

relative to the market dynamics in effect during the time the track records were achieved. But better-than-average track records are a statistical inevitability in games with many players and a large element of chance, and as such they do not by themselves provide evidence for the attainability of a sustainable investment or trading edge.

1.2 Considerations for Forecasting Models

Whether or not any given fund manager's outperformance may be attributed to skill, there is no doubt that the domain within which such professionals operate is a highly complex and specialized universe. Some understanding of this universe is helpful, and probably necessary, if we aim to operate within it autonomously in pursuit of market besting returns in spite of a core body of theory arguing for the futility of our mission. Trading strategies developed for the purpose of outperforming the market are informed (or at least ought to be informed) by an understanding of how markets work, at the mechanical level, and by forecasting models that account, perhaps implicitly, for the technical and psychological forces which move prices. The construction of a forecasting model may thus be well served to consider the motivations of market participants, the types of trading strategies and the execution mechanisms those participants employ, and the psychological and behavioral tendencies that play a large role in determining the perceptions of value and risk which drive price discovery. To the extent an understanding of these features provides us with a perspective on the dynamics underlying market behavior, this perspective can inform our decisions during model development.

Before discussing neural networks and the particulars of our forecasting model, we touch on some practical and theoretical concepts which relate to its development and to our methodological approach- or which simply provide context for our endeavor. We discuss EMH in more detail along with the related Random Walk hypothesis. We talk about some of the mechanisms, often computerized, by which securities are traded, and we touch on technical analysis as it relates to discovering psychologically meaningful price patterns. We briefly discuss Game Theory and take a related tangent into evolutionary biology in an effort to illustrate a dynamic that, if projected to markets, may offer insight into the observation that trading strategies, as they start to become profitable, tend to become ineffectual in fairly short order- only to re-emerge later with renewed viability.

1.3 Artificial Neural Networks

Artificial Neural Networks (ANN) are function approximating mathematical models which process inputs in a way that bears analogy to the how brain cells (i.e. neurons) process sensory information. Layers of neuronal nodes receive inputs via weighted connections (think parameters and coefficients) from the various input or intermediary nodes of the preceding layer and transform these into output by way of an activation function. The ability of such networks to handle non-linear relationships between the inputs without the need to specify those relationships makes them especially useful for modelling complex processes like those in play with price time series. The Multilayer

Perceptron is a common form of ANN and one we will make use of in our research. We will thus describe how ANNs and their MLP strains work in general, and we will discuss their use in market forecasting models in particular.

1.4 Problem Statement and Research Goal

The application of machine learning techniques to market forecasting has been explored by a wide variety of researchers. Artificial Neural Networks (ANN) have figured prominently in this area. We find some common shortcomings with respect to the generalizability of such research. While predictive performance using statistical error measures may generalize over multiple test intervals, return performance may not because:

1. Statistical error measures are not strongly correlated with profitability.
2. Benchmark comparisons will be more or less favorable for different test periods (vs. B&H, for example).
3. Return performance may vary with market conditions and may thus be susceptible to poor timing relative to initial deployment (a system may not be able to recover from a period of poor returns when it occurs early in deployment).
4. The underlying dynamics by which prices are generated may not remain constant over time (predictive factors or the relationships between them may change). The future may not resemble the past.

Additionally, in research where return performance is reported, we find some cases which appear to apply trading rules developed after observing the behavior of a predictor

upon the primary test set. We believe this raises questions of bias being built into some reported results with respect to return performance.

We argue these concerns provide reason for skepticism relative to ANN research claiming to undermine EMH. Some of these previous studies may very well be indicative of sustainable trading advantages obtained through the use of ANNs, but such claims would be stronger where none of the issues above is present. Consequently, the goal of this thesis is to test the ability of ANNs to provide trading signals that produce market besting returns reliably and portably, and thus pose a challenge to EMH, using a methodology which:

1 Measures results primarily in terms of dollar-valued returns rather than statistical error measures.

2 Forecloses the possibility of results that are the product of a fortuitous sequence of predictive signals projected onto a fixed-interval test period by using uniquely initialized MLPs trained on up-to-date data prior to each set of predictions.

3 Seeks to demonstrate the repeatability of our results by performing multiple tests upon the same intervals, with the prediction sequence of each run resulting from an MLP ensemble that is dynamically, and thus uniquely, trained over each run.

4 Attempts to show that return performance, rather than performance relative to simple error measures, can be generalized to multiple test sets over various date ranges.

5 Ensures trading results are not biased due to selectively applying rules determining when or how our ANN's output, or signal, will be considered actionable

subsequent to having observed the relationship between the price and prediction (signal) sequences over the test period.

To the extent we can demonstrate success with ANNs with such an approach, it may provide a more convincing argument for the use of ANNs in market trading decisions and a more rigorous approach for conducting this research in the future. However, we will argue that a failure to do so may be a more consequential outcome. While the use of ANNs should not be discounted as tools to guide trading behavior due to the results of one study, it is reasonable to argue that the higher bar set here should be met for claims against EMH to be persuasive. Accordingly, the contribution of this research will have more to do with methodology than with elegant algorithms or idiosyncratic ANN implementations, but some energy will be expended refining the model in order to compete with the results reported by previous research. We argue that, regardless of any limitations inherent in our implementation, past and future claims of success with ANNs in obtaining above market returns will be strengthened by successful studies applying methodology similar to that employed here.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Financial Theory and Trading Behavior

The trading of financial securities is done with a great many approaches and is aided by tools and methods borrowed from a large number of disciplines. Machine Learning and Artificial Intelligence are commonly applied to the discovery and improvement of trading algorithms. Wall Street hires a large number PhDs in mathematics, physics, computer science, statistics and finance [Quants13] to develop and refine these approaches. Large institutions have mandates to buy and sell large amounts of assets, and brokerage traders are tasked with executing these orders at the most favorable terms achievable. Increasingly, brokers accomplish this task with the use of algorithms implemented on automatic trading systems. Efforts to divine the dynamics governing price discovery and speculative behavior in highly liquid markets are central to much of financial theory, and these may be approached with reference to many disciplines. If our goal is simply to apply ANNs to the production of broad market forecasts, then the day-to-day dynamics of trading activity might be considered superfluous to our endeavor. But if it is our intention is to employ machine learning techniques to generate real-time trading signals, then some understanding of both financial theory and trading mechanics is in order.

We discuss some of these issues here to provide context for our task, and we reference this context to inform the construction of our forecasting model and trading strategy as we go forward.

2.1.1 Technical Analysis, Random Walks and EMH

Technical Analysis (TA) is an approach to investing that analyzes the statistics generated by market activity with an eye toward finding patterns useful for choosing future investments. While many investors and traders use TA tools in combination with fundamental information, TA is agnostic with respect to such fundamental data. Rather, TA attempts to discern supply-demand patterns from past price-volume data, and to infer likely future price directions, often contingent upon how prices progress relative to key technical hurdles. Various mathematical indicators coupled with charting and visualization tools may be transposed onto price charts for purposes of divining useful patterns. Moving averages, Candlestick charts, trend-lines and Bollinger Bands are just a few amongst a great many such indicators. The value of any security, however, is of no concern for TA. Rather, it is the behavior represented in the price charts that provides indications about future prices. See [Murphy99] for and extended explanation of TA techniques.

Utterly incongruous with TA is the school of thought in finance which holds that fluctuations of asset prices are, for all practical purposes, merely random sequences. The Random Walk Hypothesis (RWH), much debated in the latter 20th century thanks to its

popularization in [Malkiel73], can be traced as far back as the mid-19th century [Regnault63]. More often credited is [Bachelier00], upon which the modern conceptualization is predicated. Simply stated, RWH holds that market prices move with the same practical indeterminism as particles exhibiting Brownian Motion and are thus observationally equivalent to a random series. Speculation, by this view, is but a feeble enterprise.

Less severe (but only slightly so) for the speculator's endeavor is the Efficient Market Hypothesis (EMH). Propounded by [Fama65], EMH states that modern markets are efficient and, being so, incorporate all information relevant to an asset's price immediately, thus leaving no room for speculators to gain returns exceeding those of the overall market. While the strong form of the hypothesis implies RWH, the weakest form allows that asymmetries in fundamental information may occasionally provide excess returns. However, even this weak form holds there are no serial correlations in the time series represented by asset prices. Future prices, by this view, are entirely determined by information not contained in previous prices. It follows that we cannot systematically exploit past prices to gain an edge on the future.

Obviously, we cannot hold out EMH as our pricing model, on the one hand, and claim to apply technical analysis profitably, on the other. Nevertheless, both are intuitive. Assets do seem to follow predictable patterns at times, and traders have gained legendary status by exploiting technical patterns in spectacular fashion [Faith07]. Yet, markets do seem

to incorporate information very, very quickly; and our technical tools fail us utterly all too often.

Absent some reconciling logical framework, it seems we must reject either EMH or the tools of TA if we are to lay claim to a coherent view of market behavior. Perhaps technical analysis is but another example of information being assimilated by efficient markets? TA may merely provide tools for describing the dynamics of past price histories, while the market's assimilation of TA's products renders these tools impotent to discern the new dynamics created by their mass digestion. The results from [LeBaron92] suggest as much. These indicate that, while nonlinear price regularities seem to exist, they are unstable over time. [Chen97] Also found such regularities by applying Genetic Programming, but the authors noted the cost of discovery likely limited profitable exploitation.

2.1.2 Program Trading

Program trading represents a broad set of computer executed trading strategies employed by financial firms and speculators. Perhaps the best publicized, if not infamous, kind of program trading is High Frequency Trading (HFT). Firms executing HFT strategies aim to take advantage of informational asymmetries brought about by speed advantages gained from highly optimized hardware-software systems that are co-located with the exchanges they trade on. Such systems provide multi-millisecond visibility advantages to order books, allowing for instantaneous profits to be made by gaming both sides of the

bid/offer spread. Put another way, HFT systems make their living by knowing what buyers and sellers are willing to do (pay/accept) before either is exposed to this same information from would-be counterparties. This kind of algorithmic trading essentially amounts to high-tech, rapid-fire arbitrage. Lewis [Lewis14] provides a detailed, albeit non-academic, account of this type of trading activity.

Long standing and more innocuous forms of program trading are essentially automated versions of traditional brokerage strategies for buying and selling large blocks of shares. These strategies are typically what Wall Street people mean when they refer to algorithmic trading, and they are designed to minimize both the market impact (unfavorable price changes resulting from trading activity) and the transaction costs associated with the execution of larger orders. Volume Weighted Average Price (VWAP) and Percentage of Volume (POV) are common benchmarks the algorithms attempt to beat with various implementations [Johnson10]. Still other types of programming trading are quantitative trading, where participants try to predict short term price moves to obtain quick profits on transient market moves, and statistical arbitrage strategies which spot short to medium term anomalies in the price ratios of correlated securities.

Perhaps the most important thing to understand about all of this effort toward profitable price prediction is that, for a given security in a given market, opportunities to exploit recent patterns require early awareness and are constrained by finite liquidity. As formerly profitable patterns are discovered and exploited by more market participants,

those opportunities will cease to be profitable. Knowledge essentially undermines itself as resulting behaviors cause patterns to cease to mean what they just meant. Patterns which formerly served as reliable buy signals come to be exploited by sellers, and sell signals are likewise then exploited by buyers, and then this situation breaks down in turn. Yet there will inevitably be discoverable, and thus temporarily exploitable, patterns reflecting this new situation- so long as this new situation holds. This counterbalancing dynamism, we argue, constrains our ability to generalize about any fixed system's predictive abilities on a continuing basis. Static price forecasts produced by predictive models developed (or trained) over fixed time intervals would seem inappropriate tools if one views the trading environment in this way. This characterization of price behavior relies on informal observation more than theory or empirical testing, but support for it can be found in [Faith07, Lempérière14 & Clark12].

These dynamics are of critical importance for anyone attempting to trade in the equity markets based on signals provided by forecasting models. Underlying fundamental conditions will likely drive prices over the intermediate to long term, but the multifarious and ever present pursuit of a technical trading edge by so many market participants may create short term price behaviors that confuse forecasting models into loss-making trade signals. If past price patterns proved not to be reliably exploitable as a consequence of this condition, well, that is exactly what EMH proponents have been trying to tell us. A forecasting model powerful enough to undermine EMH would have to be highly adaptable to ever changing conditions.

2.1.3 Game Theory for Lizards

Game Theory provides a way to think about the ways in which self-interested participants interact strategically with one another. As such, it offers insights for a number of disciplines including economics, psychology, computer science, political science and biology, to name a few. The root of Game Theory's academic tree stems from Jon Von Neumann's work on zero-sum-games [Von Neumann44], though discussions of matters with which it is concerned can be found throughout written history. With respect to economics and finance, Game Theory typically thinks of individuals as rational, self-interested agents attempting to maximize some sort of utility function. Games (defined settings where behavioral choices determine the results of individual utility functions) may be zero-sum, or they may reward cooperation. Markets, of course, exhibit zero-sum games between participants and are amenable to such models. Because the applicability of game theory crosses so many domains, it provides rich metaphorical soil for conceptualizing the dynamics of many types of systems. Tilling that soil, we will look to a game theoretic view of the plight of some peculiar lizards as a means to illustrate a view of market behavior which informs our methodological approach.

The notion of an Evolutionary Stable Strategy (ESS), a product of the application of Game Theory to the evolution of behavior, refers to those strategies employed by species which, once adopted by all members, cannot be invaded or overrun by an initially rare outside strategy. [Sinervo96] Studied territorial and sexual selection patterns of male

side-blotched lizards. Three distinct phenotypes of male lizards compete for female resources in this species. Males with orange coloring about their throats are physically dominant over all others and control the largest ranges of territory. Blue throated males dominate yellow throated males but control smaller territories than orange throated members. Males with yellow stripes on their throats are known as "sneakers" and do not control territories. Rather, they look like receptive females bearing the same markings and engage in a subterfuge for reproductive advantage. Each "morph", then, employs a distinct strategy for procuring females relative to the other two morphs. Statistics for the relative frequencies of morphs over a six-year period demonstrated that no morph maintained an ESS, as the frequencies of morphs fluctuated dramatically from one year to the next. In particular, the authors observed every morph was vulnerable when it was prevalent, and that the morph least represented in any given year always faired best the following year.

It seems too clever by half to extrapolate a theory of markets from a slice of biology, and we are not proposing one[1]. But then game theory is concerned with dynamics rather than domains, and the dynamics here comport with our observations on the fortunes of market strategies. We will not stretch our arguably tangential metaphor much further, except to say the idea that strategic success is but a prelude to strategic futility, that this is also true in the reverse, and that each position comes back around in a roughly (if very roughly) cyclical fashion; well, this is our informal model for understanding the alternating success and failure of market strategies. We might say that the thing which EMH

[1] But see (Soros, 2003). Mr. Soros is concerned with human behavior rather than the colors of lizards' throats, but similar implications can be said to follow from his view of markets.

proscribes, and the thing we must prove to exist if we wish to controvert it, is an ESS where the market serves as our environment and profitability as our success measure.

This line of thought will have implications for the design of our ANN based system. As we will discuss in more detail in the next chapter, most ANNs are trained over a fixed in-sample period. If we view market activity as evolving in a loosely cyclical fashion, where it is not just price trends but also the suitability of strategies best used to exploit those trends that are in constant flux, then confining the training of our model to a fixed period would seem an inadequate approach to predicting prices. Consequently, our model will be trained in an iterative fashion not often employed in the literature.

2.2 Artificial Neural Networks

The simplest artificial neural network is the single-layer perceptron, popularized by Frank Rosenblatt in the early 1960's [Rosenblatt58]. A single layer of output nodes (or neurons) is fed input data via weighted connections. These output neurons fire when the sum of the products of the inputs and weights are above a specified threshold. Raw data are converted into a set of feature activations and, through training, perceptrons learn to weight each feature such that the weights represent how much evidence a feature provides in favor or against the current input being an example of the pattern or value we wish to recognize, or predict, via the outputs. As these types of ANNs are severely limited, in particular by their inability to discover patterns which are not linearly

separable [Minsky69], we do not discuss training algorithms with respect to the single-layer perceptron.

The most common type of ANN is the multilayer perceptron (MLP). These are Feed-Forward Networks (FFN), meaning they process information in one direction. The first layer in these networks is the input layer and the last layer is the output layer. These networks compute a series of transformations on the data vectors from the input layer via one or more hidden layers of neurons where data arrive by directed, weighted connections and proceeds transformed to the output layer (perhaps via additional hidden layers). With the exception of the input layer, each node (or neuron) processes inputs via an activation function such that the activities of neurons in each layer are non-linear functions of the activities in the preceding layer. [Hornik89] demonstrated that an MLP with a single hidden layer is capable of approximating any continuous function. Because of the general ubiquity of MLPs and their centrality to our research, we occasionally appear to interchange the terms MLP and ANN in this document. However, the term ANN should be considered to refer to Neural Networks in the more general sense.

Recurrent ANNs allow for directed cycles in their connection graph. This distinguishes them from FFNs, which allow no such cycles. These cycles may capture temporal relationships and thus may provide for more complex descriptions. Recurrent networks are much more challenging to train than feed-forward networks. They are, however, more biologically realistic, and the element of memory introduced by recurrent cycles make them potentially more powerful. They are also a very natural way to model

sequential data. Two of the most common recurrent structures are Elman networks and Jordan networks [Elman90, Jordan86].

Many more variations of ANNs exist (probabilistic neural networks, or PNNs, are another common form), and hybrid systems combining neural networks with fuzzy sets, genetic algorithms and all manner of machine learning exotica are quite common [e.g., Asadi12, de Oliveira13, Fang14].

The method by which ANNs are trained must be appropriate to the specific ANN structure and usually takes account of efficiency considerations. The standard back-propagation algorithm used to train MLPs employs gradient-descent optimization to find the local minima of the error function, and the weights of the connections are adjusted with each instance encountered by the amount of a specified learning rate parameter. This is accomplished over multiple (often very many) iterations, or epochs, over the entire training set. The algorithm may be (and usually is) extended with a momentum parameter that helps smooth out some of the oscillations of the gradient which can slow down learning. While a higher rate of momentum can cause faster convergence, it brings with it a risk of early convergence onto local minima. It is thus common to reduce the learning rate in conjunction with using higher rates of momentum.

Other training techniques use multiple optimization procedures for faster training, as does the Levenberg-Marquardt algorithm [Levenberg44] which interpolates between back-propagation and Gauss-Newton optimization. Additionally, genetic algorithms

may be used to determine MLP weights. Resilient Backpropagation, or RPROP [Riedmiller92], uses a special update value for every neural connection, similar to the learning rate of backpropagation, which is automatically determined rather than pre-specified as a parameter (only the initial update value must be specified). RPROP has been shown to perform more efficiently than backpropagation. Because finding the right combination of learning rate and momentum can be extremely time consuming when constructing many MLPs, RPROP provides a significantly less resource intensive training option.

For an MLP with a single hidden layer, determining the appropriate number of hidden nodes is an imprecise endeavor requiring experimentation. A common heuristic is to begin by taking the mean of the nodes in the input and output layers [Heaton08].

2.3 Neural Networks and Market Forecasting

Research with ANNs in financial modeling began in earnest in the early 1990s [Franses98]. Because ANNS are capable of approximating almost any nonlinear function with arbitrarily high precision (given enough hidden nodes), they are much better than traditional linear econometric models at discovering highly complex relationships between the lagged components of many financial time series. This precision comes at some cost, however. Because ANNs are non-parametric statistical models, they do not lend themselves to parametric interpretation. They are essentially "black-box" functions that, while highly capable of discovering nonlinear relationships,

provide little information about the nature of the relationships discovered. This situation means ANNs suffer in terms of their explanatory value for specific models, and it makes for challenges with model selection- for example, determining the number hidden nodes to include or the particular transformation function(s) to be applied.

The black-box aspect of ANNs is a tolerable drawback in a forecasting model used merely as a trading tool, so long as their approximating abilities may be generalized sufficiently to future data. For a given model, however, there is no assurance of the degree to which the precision achieved in approximating an in-sample time series will extend to the future outputs of that series. The forecasting ability of ANNs thus depends on the similarity between the unknown data-generation processes (including noise) in effect during the interval in which the ANN was trained and those of the future interval for which we desire forecasts. This fact has methodological implications for our research, but here we simply remark that the dangers of both over-fitting and under-fitting a model to in-sample (training) data are serious concerns, and that the further a model extends forecasts beyond its in-sample period, the less reliable we might expect it to be as a forecasting tool.

Kuan and Liu [Kuan95] had mixed success with MLPs in predicting five exchange rates against the US Dollar. For at least some of these series they were able to demonstrate significant performance in terms of Mean Squared Prediction Error (MSPE) and sign prediction (hit rate) of MLPs over the random walk model. These MLPs, using only the lagged values of their respective time series as inputs, were simple autoregressive (AR)

models. Gencay [Gencay99] followed the above research by feeding k-nearest-neighbor time-series into a feed-forward network for the purpose of predicting spot foreign exchange rates across five currency pairs with data from January 2, 1973 to July 7, 1992. Results of this study show a 7.9% improvement in returns over the RWH model on out-of-sample data, as well as more accurate sign (direction) predictions. In [Fernandez-Rodrıguez00], technical trading rules determined by an ANN trained and tested on percentage returns of the Madrid Stock Exchange performed better than a buy-and-hold strategy in bear and stable markets, while performing worse in bull markets. The authors here follow the Gencay model in using the returns from the nine previous days as inputs to predict short term time series patterns.

It is important to point out how the latter two models differ from the former. Where Kuan and Lu were attempting to estimate the values of exchange rate time series, Gencay and Fernandez-Rodriguez et al looked at the returns of those series. These obviously produce very different regression lines, with the latter crossing between positive and negative percentages for both actual and predicted values. These signed values provide obvious trading signals for choosing between long and short (or cash) positions. Of course, it isn't necessary that that we observe the zero line as the absolute signal (though these studies do). As we discuss in Chapter 5, we can require that the signal achieve some level of magnitude beyond the zero line before changing our market position, depending on our investment or trading strategy.

Autoregressive models which consider only the lagged values of a time series for which predictions are desired might be expected to suffer from their narrow concern with the relationships between the values of the series itself. Markets do not operate in a vacuum, and their sensitivity to exogenous events is hardly a matter of debate. How and under what conditions external forces exert their influence on market prices most certainly is a matter of seemingly boundless debate amongst financial practitioners, however. ANN models, given their non-parametric structure, are not good candidates for elucidating the nature of these relationships. Their approximating abilities for nonlinear functions nevertheless make them very good candidates for discovering these relationships implicitly - even if the nature of the discoveries remain unclear. Autoregressive Models with Exogenous Inputs (ARX), which incorporate relevant external information, might thus be expected to provide significant forecasting improvements over simple AR models.

Brabazon et al provides a good example of an ARX model [Brabazon06]. Here, a range of exogenous market index values and derived indicators are used as inputs along with the last value in the series to predict values for the FTSE 100 index 5 days forward. Altogether, these variables produce an input layer with 10 nodes (plus a bias node). The greater the number of inputs, the more hidden nodes will typically be required to capture the greater number of relationships. Brabazon's MLP uses 6 hidden nodes and produces outputs in the range of [-1, 1]. This output range reflects the normalization of inputs into this same range and the use of the hyperbolic tangent function as the transformation function. Rather than rely on the predictions of a single MLP, where the initial weights

used for the backpropagation algorithm can produce very different approximation results, 25 MLPs were trained and the predictions of each were averaged to produce a single output value. De-normalized outputs having absolute values greater than 1.5% (predicted +/- return) were taken as long or short signals in the system constructed to test the usefulness of the model for trading over subsequent test periods. As expected under the hypothesis that markets are dynamic, the out-of-sample performance- measured both by standard error measures for the MLPs and by the returns of the trading system- deteriorated with each subsequent test period. The trading system did produce modest returns over buy-and-hold in the first and second test periods, however, and similar results were also found when training a single MLP with a genetic algorithm (GA), where stacking MLPs isn't necessary because using GAs avoids the problem of poor weight initialization.

Chapter 3

METHODOLOGICAL CONCERNS FOR FORECASTING RETURNS WITH ANNs

A review of the literature on the use of ANNs and, more specifically, MLPs for forecasting price trends in financial markets suggests these models can provide useful information to financial practitioners. Yet, despite a number of studies reporting better than benchmark returns from trading systems constructed upon these structures, a closer look at this research raises suspicions as to the ability of MLP based forecasts to provide reliably profitable trading signals. We discuss the reasons for such suspicions in the pages that follow. First, we construct a simple example model for purposes of illustration.

3.1 An Example Model

Let us construct an MLP for predicting index closing values for the S&P 500, the Dow Jones Industrial Average, and the Nasdaq 100 index values[2]. Our example MLP is of the ARX statistical variety and uses the lagged 10 closing values of each index along with the lagged 10 closing values of the Prime (Federal Funds) interest rate. This gives us a total of 40 input neurons. Remembering that our purpose here is demonstrative rather than formal, we use a rather large, non-optimized structure with 2 hidden layers of 41

[2] The code for this example MLP was provided by http://www.codeproject.com/Articles/175777/Financial-predictor-via-neural-network. Trading statistics were the product of our calculations.

nodes each and an output layer of 4 nodes (1 for each series), each of which produces continuous outputs. This gives us a structure of 40-41-41-4 (we ignore the interest rate predictions, however). A rough visualization of this MLP can be seen in Figure 1.

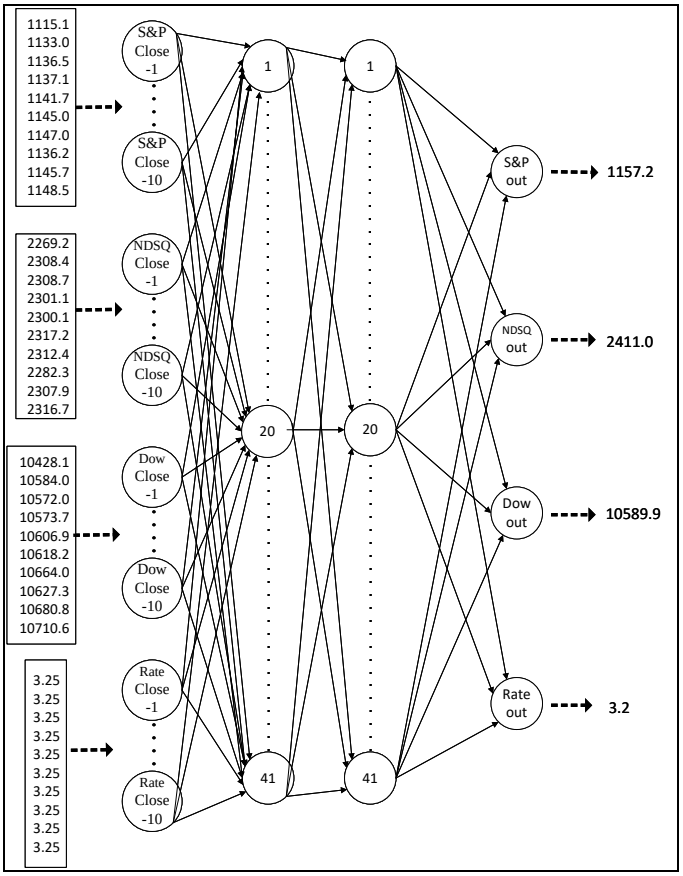

Figure 1: Example MLP (Bias nodes not shown)

We train our MLP on daily closing index values from January 1, 1990 through January 1, 2010, stopping training after 1,000 epochs. We make predictions for each index from January 2, 2010 through May 5, 2011 (based on the previous 10 values in the series), for a total of 81 predictions for each index. Graphs of actual and predicted values for each of the 3 stock indices are shown in Figure 2 (we ignore the Federal Funds Rate predictions).

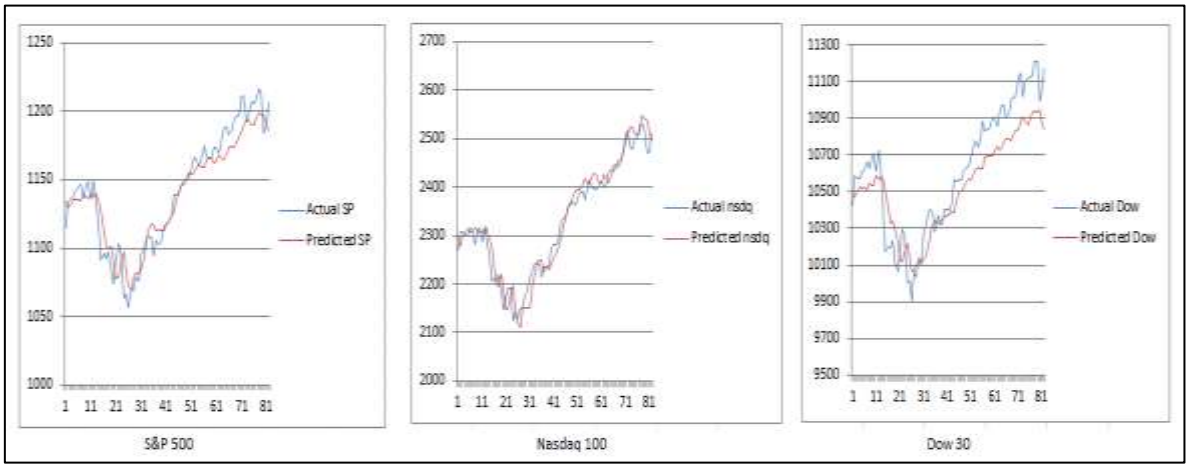

Figure 2: Graph of actual vs. predicted values for example MLP

The results of this example model now in hand, we reference it in the context of specific methodological concerns.

3.2 Measuring Forecast Performance

As can be seen in Figure 2, worries that the approximating ability of our MLP will suffer when extended to out-of-sample data do not appear to be warranted. Indeed, each of our forecasts has a Pearson Product Moment Correlation Coefficient (PPMCC) above 0.95 and the Root Mean Squared Errors of our predictions range from 1.05% to 1.50%, depending on the index and its value when measured.

Given how well our MLP approximates future values, would we have profited by trading based on the signals provided over the out-of-sample period? After our initial MLP training, if at the end of each trading day during the out-of-sample period we had bought

or sold- based on the price predictions of our example MLP- any of a half dozen securities whose values mirror the S&P 500 index, we would have outperformed the S&P by more than 60% over a period of less than 4 months.

That kind of performance is impressive. It is also ephemeral. We need only look to our example MLP's predictions for the other 2 indices to see just how unreliable the above trading results likely are. Despite sporting PPMCCs of above 0.95 and RMSE values comparable to those for the S&P index, had we traded securities reflecting the NASDAQ and Dow averages we would have underperformed these indices by more than 59% and 81%, respectively.

The idea that standard error measures may be inappropriate, or at least insufficient, for measuring the ability of approximating systems to produce profitable trade signals is not new. Diebold and Mariano point out the economic loss functions aren't amenable to textbook error measures like Mean Squared Prediction Error (MSPE) and the like [Diebold12]. This is because loss functions in economics, particularly as they pertain to investment decisions, are typically non-Gaussian, non-zero mean and contain both serial and contemporaneous correlations. Pearson and Timmermann demonstrated these measures are not strongly related to profitable trading [Pesaron92]. Yet a surprising amount of research cites these error measures exclusively as principal evidence for the profitability of exploiting ANNs for this purpose. We see this in Birgul [Birgul03] where multiple MLPs are said to better predict values for the Istanbul Stock Exchange than other methods, in Constantinou et al [Constantinou06] which looks at using ANNs for

predicting Cyprus Stock Exchange values, and with Jaruszewicz and Mańdziuk [Jaruszewicz04] where next day predictions are made for the Japanese NIKKEI index. The ANNs used in these studies do appear to outperform when approximating these markets, but the relationship between approximation capacity and profitable trading remains tenuous. Kanas and Yannopoulos [Kanas01] provide more convincing evidence of ANN forecasting superiority by demonstrating their relative forecast outperformance, using error measures designed by Diebold and Mariano [Diebold12], which account for the messiness inherent to economic loss functions. Yet, even here, the lack of any attempt to demonstrate generalizable, above benchmark returns leaves us less than persuaded.

Leitch and Tanner [Leitch91] argue that a system's ability to forecast the direction or sign of market returns, sometimes referred to as hit rate, is more closely related to profitability than traditional error measures. This is most certainly true, but it also has limitations for this purpose. This is because the determining factor for profitability is not the ratio of hits to misses but rather the magnitude of gains on hits relative to the magnitude of losses on misses. Indeed, profitable systems need not have a high hit rate at all, and some famous trend following systems are known to have had hit rates of less than 20% [Faith07]. So long as the many losses are minimal and the relatively few successes are very large, such systems can be extremely profitable. But systems attempting to achieve many small gains are indeed dependent on higher hit rates for profitability, as hits must outnumber misses when average gains and average losses per trade are near equal. However, such systems suffer during times of low volatility when

their returns are less likely to compensate for the costs of active trading. They are also vulnerable to extreme moves, as small returns amassed over many trades can be wiped out by one extreme move in the wrong direction. Nonetheless, hit rate is a useful measure, and we see it cited in the literature frequently [e.g., Mizuno98, Pan05, Neto10, Tsai09].

3.3 Issues with Out-of-sample Testing

Though the returns produced for the S&P 500 failed to generalize to the other two indices, we might postulate that our example MLP's trading performance on the S&P 500 has validity for reasons internal to the data generation process producing this index series. After all, there are 500 securities composing this index (as opposed to 30 and 100 for the Dow and NASDAQ 100), and we can imagine that this larger, broader composition of securities might produce time series vectors which are more reliably predictive of future price direction and/or the magnitude of price changes, once they are processed by our MLP. Unfortunately, that is not the case here. Three subsequent training and testing runs produced predictions and trading returns for the S&P 500 that were comparably bad or worse than those obtained for the Dow and NASDAQ indices on the first run. Our very first result set, in terms of trading performance, appears to have been an anomaly. Table 1 provides the results of 5 additional runs for each index and compares the average returns produced by following the predictions produced by these runs to the Buy & Hold strategy over the same period. As with the previous results, PPMCC statistics suggest the MLP is provides very good price level approximations.

Our approximations, however, appear to be too imprecise to allow us to capitalize on them in order to reliably beat the B&H return from these indexes over the period.

	Returns From Example MLP (5 Runs)								
	Ex. S&P Ret. (pts)	Ex. S&P Ret. (%)	Ex. S&P PPMCC	Ex. NDSQ Ret. (pts)	Ex. NDSQ Ret. (%)	Ex. NDSQ PPMCC	Ex. Dow Ret. (pts)	Ex. Dow Ret. (%)	Ex. Dow PPMCC
	63.12	5.66	0.9309	201.55	8.88	0.9314	324.19	3.11	0.9322
	103.26	9.26	0.9620	399.83	17.62	0.9285	131.01	1.26	0.9577
	-110.14	-9.88	0.9589	-58.41	-2.57	0.9715	-65.05	-0.62	0.9584
	21.32	1.91	0.9665	-114.29	-5.04	0.9665	389.63	3.74	0.9665
	21.20	1.90	0.9648	-15.49	-0.68	0.9747	-538.85	-5.17	0.9406
Ex. Avg.	19.75	1.77	0.96	82.64	3.64	0.95	48.19	0.46	0.95
B&H	91.68	8.22		242.77	10.70		739.27	7.09	
vs. B&H	(71.93)	(6.45)		(160.13)	(7.06)		(691.08)	(6.63)	

Table 1: 5 Runs of Example MLP with B&H Comparisons

That we can produce these exceptional trading results, however specious, speaks to another methodological concern when investigating the ability of ANNs to provide profitable trading signals. For any finite out-of-sample test period, we are likely to find a profitable set of predictions- given enough parameter tweaks and testing runs, or with just a little luck. The autoregressive nature of ANNs will usually get the predictions "in the ballpark"- but then so will a Monte Carlo simulator. Thus, finding one or more profitable sets of predictions over a given period provides little support for a claim that any single, uniquely trained MLP will generalize to profitable predictions going forward. It just means that we can be confident in our ability to back-fit profitable returns to the recent future- so to speak.

Yet, back-fitting profits to futures' past seems to be a prevalent approach for researchers proclaiming the death of EMH at the hands of machine learning techniques, in general,

and with ANN techniques in particular. Fortunately, most researchers in this area do not make such grandiose claims. Nevertheless, there remains the essential problem of how well we can expect excess returns produced over one period (or date range) to generalize to another.

The total returns of any trading or investment system, no matter how effective, are highly dependent on when trades (or investments) are made. Specifically, the intermediate and possibly long-term returns on a given purchase (or short sale) of shares will depend largely on whether that purchase takes place at the bottom or top of a market boom/bust cycle, or somewhere between. This is a mere function of the share price, and thus the number of shares bought or sold, at the time of investment.

This is obviously true for the buy-and-hold (B&H) investment strategy. For example, purchase of an S&P 500 tracking security in December 31, 2000 would have yielded an investor a return of -0.019% by January 3 of 2013 (before dividend reinvestments), whereas an equivalent dollar investment would have yielded that same investor a total of 73% had she bought her shares on March 3, 2003. Comparisons of a strategy's returns to B&H are thus heavily fraught, for such returns may be flat or negative for an extended period, thus inflating the alternate strategy's performance by comparison.

For strategies other than B&H, the consequences of poor timing may be even more dramatic. Any strategy that attempts to 'time the market' with long and short trades, for example, will incur periods of large run-ups and large drawdowns, where it is either right

or wrong for a relatively extended period, even if that strategy's long-term success rate is stable. Should either be the case early in a strategy's deployment, short-term returns may be inflated or suppressed enough to impact the returns for the entire period under consideration. We can't know how a system would have performed had the situation been different unless we test it against periods where it was different.

Crucial to the establishment of a systems ability to generalize is its ability to repeat good performance given slight variations in initial parameters and the datasets tested against. While a statically trained system with fixed connection weights will always produce the same outputs if given the same inputs, the system must be robust to slight changes to the training and testing datasets if it is to be expected generalize to new data. For example, will our return performance maintain its superiority over B&H if we add or drop portions of the training data, or if we shift forward or back the date of initial testing/deployment? Will our finely tuned model parameters, perhaps ideal for our test dataset, be suited to future datasets? Such variants must be tested against to be sure we haven't merely stumbled upon a profitable but arbitrary prediction sequence.

Yet, such variability is rarely tested against. If you have trained an ANN and employed it profitably over a finite out-of-sample period then, if the profitability of that ANN's predictions is a reliable phenomenon, that profitability ought to be reproducible on subsequent training and testing runs over slightly modified training and testing intervals, and over the same datasets using alternative initialization parameters. Every run need not be better than our benchmark, and we might expect some runs to be downright losers; but

we ought to be able to beat our benchmark on average- and to at least beat it more than once. In addition, if we wish to have confidence that our thus validated system is not somehow reliant on features particular to our choice of training and testing intervals, this system and its associated profitability should be at least somewhat generalizable to other, distinct intervals.

We are unaware of research demonstrating that profits produced by MLP generated signals are consistently reproducible or temporally portable in this way. Testing this hypothesis is a key goal of this thesis.

3.4 Rules Bias

Research on the profitability of ANN based predictive systems often applies very simple trading rules. Like our example MLP, many systems attempt to predict returns for the next day and simply enter the market, long or short[3], based on the value of the ANN signal. Other systems, however, have more elaborate criteria for taking or changing market positions. These more elaborate sets of rules require extra scrutiny when testing them against historical data. In particular, we need to ensure that the trading rules are not fitted to suit our forecasting signal's behavior over the out-of-sample period.

[3]Short selling is a mechanism by which market participants can profit from a security's decline in price. Shares are borrowed (usually in an automated fashion from your broker's inventory) and then sold at the market price with the expectation of repurchasing them later at a lower price. Should the share price rise, the short position loses money until the shares are repurchased.

A strategy developed after observing the peculiarities of a predictive signal over a given period, and then tested against that period, will tend to imbue one's results with a particularly pernicious kind of bias. However sound the methodologies producing the predictive signal, however accurate the signal may be otherwise, returns reported using a trading strategy tailored to suit a signal's behavior over the test period will be highly suspicious. If this is done, there is no reason to think those rules will perform similarly upon future data to which they were not similarly tailored.

This is not to say a trading strategy should not take into account the behavior of the ANN signal, relative to a post-training dataset, in order to derive good trading rules. But data used for developing the rules of a trading strategy are not appropriately incorporated into the reporting of the results of that strategy. Rather, valid trading results require that the rules be defined a priori, or that the results are computed using out-of-sample datasets subsequent to those upon which the rules were developed, lest we confuse a talent for back-fitting one curve to another for the precision of our predictor. It is sometimes hard to know the degree to which this bias is incorporated into specific studies employing complex strategies, but an absence of explicit safeguards against it suggests its presence. This is the case with [Brabazon06], where the basis for setting a 1.5% (absolute value) predicted return threshold criteria for taking positions in either direction is not made clear. In other cases, this bias is introduced overtly (if perhaps unknowingly), with trading rules being developed with a direct view to the out-of-sample data [Kuo98]. We will take explicit steps to minimize this type of bias

Chapter 4

REQUIREMENTS AND STRATEGIC GOALS

4.1 An Overview of Requirements

EMH says nothing about the ability of forecasters to produce quality approximation models and, in weak form, does not preclude the occasional achievement of above-market returns due to some intermittent informational advantage, perhaps gained by use of such models[4]. Rather, EMH precludes any such advantage from being sustained and thus systematically exploited over time. As we attempted to show in the last chapter, research with ANN based models appears to have yet to undercut this claim convincingly. However, this failure doesn't necessarily serve to bolster EMH. This is because, given the design issues we discussed in the last chapter, ANN research with market forecasting has yet to present EMH with a frontal test- at least not one of which we are aware.

To prove an advantage is reliable, we are required to demonstrate that the advantage can be obtained with some consistency, rather than be merely discoverable via trial and error or by back-fitting trading rules to a particular predictor's performance on a test-set. Demonstrating this requires that we show more than just our system's ability to produce

[4] Heretofore, EMH should be taken to refer to the weak form exclusively, unless otherwise specified.

a profitable prediction sequence over a given time period. The difference between positive and negative returns, or between better-than or worse-than market returns, may be largely determined by only a few hit or miss predictions during an out-of-sample sequence. As we demonstrated in chapter 2, we are likely to stumble across a sequence of predictions that produce trading profits; we must show that the process by which our prediction sequences are generated is one which outperforms our benchmark reliably. That process depends not only on our model's initial structure, but also on the training of the model. The weight parameters determining the out-of-sample prediction sequence will vary from one training run to the next and, consequently, so will the quality of our predictions. But if- given the same training data and the same out-of-sample test set- the average of prediction sequences for individually trained MLPs can be shown to produce above market returns over repeated training and testing runs, then we will have closed off the possibility that our success was merely the result of weight parameters having been propitiously set over a particular training run. It follows that we should train multiple MLPs and use averaged rather than unique prediction sequences for the production of our trading signals. Brabazon employs this technique successfully [Brabazon06]. Unfortunately, the results suffer from both an acknowledged failure to demonstrate robustness to time and from the apparent introduction of rules bias[5].

Even if we can show that the averaged output of our MLPs provides reliably profitable prediction sequences for a given out-of-sample period, this profitability may nevertheless

[5]Brabazon does not make any claims as to EMH, nor does he suggest his results demonstrate a sustainable trading edge. He merely demonstrates the how MLPs can be used in a trading system as well as the degradation of that system's performance over time. He goes on to suggest that a more dynamic approach is warranted.

depend on features which are unique to this period. Indeed, as Brabazon demonstrates and as we might expect from our discussion on market dynamics [Brabazon06], an MLP trained over a fixed interval tends to lose predictive power as its predictions extend further into the out-of-sample period. However, spurious outperformance resulting from a lucky or biased fit to the features of a fixed dataset can endure over multiyear out-of-sample test periods, depending on the frequency of signal generation and the distribution of returns over that period. It follows that, if we are desirous of undermining the claims of EMH, we are required to show that our system not only trains reliably well on one fixed dataset in order to predict another, but that it can maintain this reliability while incorporating new series values over time. We will want to show that, as time goes on, we can retrain our MLPs with newer data vectors and make forward predictions over more recent intervals which continue to produce above-market returns. Fundamentally, if we wish to claim our predictor can be systematically exploited for the production of above-market returns, as is required to challenge EMH, we need to demonstrate that it is robust to time.

Our last high level requirement is that our predictive system be free of any rules bias introduced by tailoring our trading signals to our test dataset. This error is very easy to make by, for example, introducing a magnitude threshold which our signal must meet to be considered actionable after observing this signal's behavior on the out-of-sample period. Should we introduce any complexity to the rules by which signals will be considered actionable, these results will be thus qualified.

4.2 Strategic Goal and Timeframe

Our strategic goal will be to beat the returns of the S&P 500 index over a 15+ year period. A common trading strategy employed in the literature is to maintain or reverse one's market position each day based on the sign of the trading signal for the next day. For example, if we are currently hold a long position in the market and our trading signal for the next day is negative, we would liquidate our long position and open a new short position at market close. Should our trading signal turn positive just before market close of the following day, we would reverse positions again; otherwise we would hold our current position. There are two obvious concerns with this approach.

First, we are asking a lot from our ANN by insisting it provide reliable predictions on a daily basis. While our ANN may do well at approximating short and medium term trends, immediate directional moves may have little to do with these. In fact, training our ANN to produce reliable predictions on a daily basis will likely come at the expense of its ability to produce reliable short to medium term predictions, as the former likely requires different sensitivities than do the latter.

Second, our trading costs are a function of our trading activity, and trading on daily signals is a fairly active strategy. Unless our ANN is exceptionally accurate, we are unlikely to beat the market after trading costs are considered. This will be doubly true during times of low volatility where directional moves are small and compensate us even

less for the cost of capitalizing on them. Our system comports with our broader strategic goal, and in so doing is be better served by being only moderately active.

We pursue our longer-term trading goal by attempting to profit from significant market downturns by shorting the market at opportune times. We attempt to do so without sacrificing the bulk of returns provided by long-term upward trends. Achieving this goal requires not only that we short the market at auspicious times, but that we re-enter long positions after market pullbacks in time to catch the largest portion of the next upward move. Simple as this may sound, it is hardly that. Every investment bank, hedge fund and active speculator participating in our market of choice would have attempted to accomplish this very thing in real-time during any historical period we might study. Few did, and few do with long-term regularity; there is indeed reason to be skeptical about the likelihood of demonstrating such ability. Nevertheless, profiting from both significant downward trends along with upward market trends is the broad strategic goal that guides the timeframes for which we make predictions with our ANN. Consequently, we measure our timeframe in weeks rather than days.

Chapter 5

METHODOLOGY AND MODEL

5.1 Machine Learning Framework

Encog is a machine learning framework which provides a large variety of functionality for machine learning projects [Heaton08]. It is free and open-source software, and the source-code can be obtained in either C# or Java. We use the C# source code here.

While Encog provides a great deal of functionality for neural networks, our use of the framework is confined to functions for building and training MLPs. Functionality related to normalization, file manipulation and windowing, while provided by Encog, is developed from scratch to suit the specific needs of this project.

5.2 Testing Regimes

We implement two testing regimes, Dynamic Training & Testing (DTT) and another, similar design which employs what we call Dynamic Validation (DV). Each methodology makes extensive use of windowing techniques common to time series data.

Windowing is used for both testing and training fixed architecture MLPs in our models. The giving of distinct names to our methods here should not be taken as a claim regarding originality.

5.2.1 Dynamic Training & Testing (DTT)

As we've specified, DTT rolls the data windows forward for both training sets and test sets such that newer predictions are consistently produced by MLPs trained on data immediately preceding the test set, thus allowing the model to adapt to changes in underlying market dynamics. While we can roll the training interval forward one data point at a time, thus producing prediction test sets of size one, efficiency concerns and preliminary work suggest we are better served by allowing each trained ensemble of MLPs to make multiple predictions. Figure 3 provides a visualization of how the testing and training periods are rolled forward.

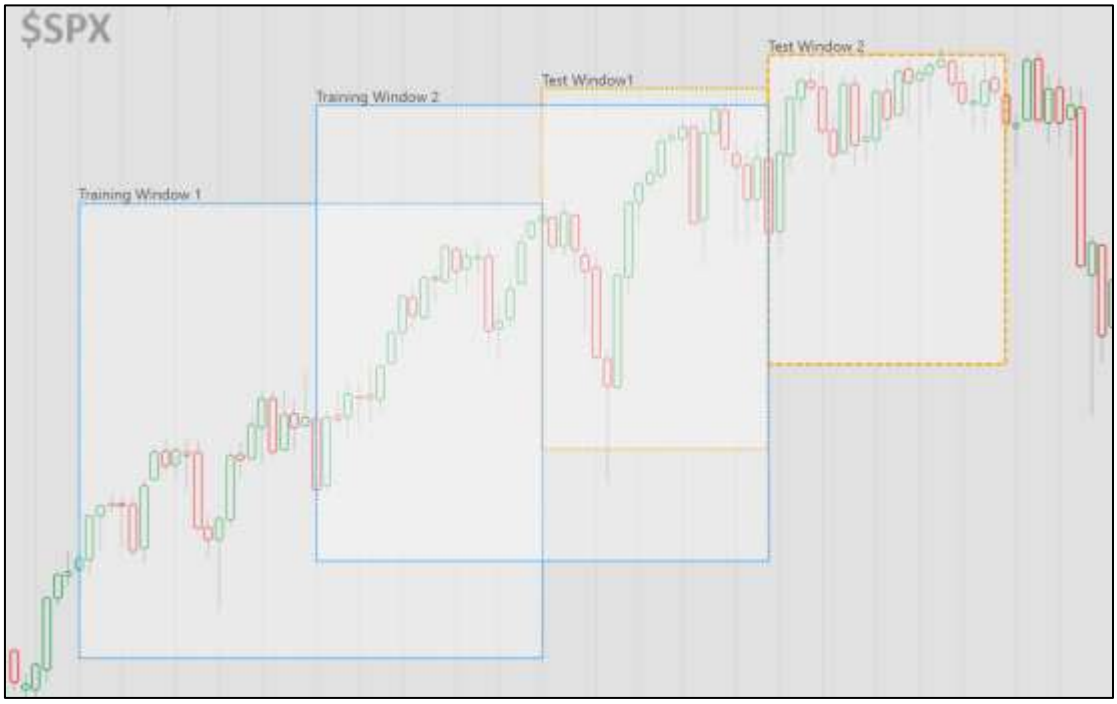

Figure 3: DTT Training Sets and Test Sets

Figure 3 shows two training sets and two test sets based on weekly data where each test set is 25 weeks long. The training sets shown here are 50 weeks long, though these may be either longer or shorter so long as they slide forward a number of periods equal to those of the test sets, which themselves may be varied between tests but remain fixed for all runs within a test (for our tests, all training sets are 25 weeks, which is also the length as the test sets).

To illustrate, the ensemble trained on the data from training window 1 is tested against the inputs and forward returns from test window 1. The training set then rolls forward a number of periods equal to the size of each test set, becoming training set 2, and once a new MLP ensemble is trained on this new data it makes predictions for 5-week forward

returns using the data derived from the period covered by test window 2 as input. The actual returns (outcomes) for the S&P 500 index upon which the MLPs are trained and tested against are determined at runtime by jumping five records ahead in the dataset and calculating the percentage change since the date of input[6].

5.2.2 Dynamic Validation (DV)

Dynamic Validation works similarly to DTT in the way training and test windows roll forward, but rather than train one MLP ensemble, we split the training set into fifths and train five ensembles. We then test these against a validation set immediately following (plus the number of forward periods predicted) the test set, and we choose the ensemble with the best hit record- in terms of predicting market direction on the validation set- as the ensemble to make predictions on the test set. Figure 4 provides a visualization of DV for a single test window.

[6]The training and test windows are shown here as being directly adjacent to one another. Programmatically, there is, and must be, a gap between these windows equal to the number of weeks forward for which returns are predicted in order not to bias the latter part of training with outcomes from the first part of the test period. As these gap weeks are always incorporated into the next training set, we elected not to display them to avoid any unnecessary confusion. This is also true for Dynamic Validation.

Figure 4: Dynamic Validation Training and Testing Intervals

Figure 4 assumes that the fifth ensemble performed best on the validation set, and the arrow labeled “Best Fit” denotes that this ensemble has been selected to make predictions over the next test interval. As with Figure 3, the intervals over which each ensemble is trained may contain more or fewer periods than the validation and testing intervals, but we always roll forward a number of periods equal to those of both the validation and test sets.

Note that our validation method is distinct from standard 5-fold cross-validation in several ways. We do not scramble the order of input vectors as is common with validation; nor are we validating single partitions of the validation set on the remainder of the validation set. This is partly because our input vectors contain temporally sensitive indicators. In addition, in order to maintain the flexibility to modify the number of periods forward for which we make predictions (during preliminary testing), the actual

outcome associated with any input vector during training is determined only when our program starts running. Scrambling the vector ordering would present a significant programming hurdle given our program's design (Encog provides functionality for this, but it is incompatible with both our design and our intention). However, the primary reason for using this method is to take advantage of potential serial correlations by choosing the MLP ensemble best suited to making predictions for the validation window immediately preceding the test window. Of course, should the return pattern within a validation window be distinctly different in character from that of the test window, we can expect to get poor results for predictions made over that test window. Figure 4 is a good example of such a case.

A final distinction of this validation process results from the criteria for determining the best performing ensemble. As with DTT, each ensemble is trained to minimize overall error. However, ensembles are validated according to their hit rate performance. While validation based on RMSE would be a reasonable choice, we have made the argument previously that maximizing directional accuracy may be preferable to minimizing the magnitude of predicted error. A plausible consequence of this decision is that predicted directional accuracy (equivalent to hit rate) may improve, even if correlations between predicted and actual magnitude changes decrease.

5.3 Prediction Target

Predictions are made for the 5-week percentage change (5-week delta) of the S&P 500 index, rather than for actual series values. Percentage change corresponds to the

percentage return obtained from having bought a single share of the index. For short sales, the sign of the return is simply reversed. As mentioned in chapter 2, predicting return percentages provides a simple directional signal based on the sign of MLP output. If desired, a magnitude threshold may be applied to MLP signals (outputs). In this case, an MLP ensemble's (continuous) output is required to reach a pre-specified magnitude above or below zero before a change in trading position is triggered. These features make return percentage the preferred prediction target here.

Of course, it is important to guard against rules bias should any complexity be introduced to the trading rules. Using zero as the cutoff for MLP output to determine long and short trading decisions eliminates the potential for rules bias, and this is the cutoff used for the first round of tests. Because, over time, upward market moves tend to significantly outnumber downward moves, it may be useful to apply a threshold for taking short positions. For one round of tests, a threshold of -1.0% is introduced for taking short positions such that in these tests we remain in, or reverse into, a long market position (betting the market will go up) when a negative prediction fails to surpass the threshold. Setting this threshold in advance provides some cover from biasing results via post-test manipulations. Any positive or negative affect on P&L performance should be taken as suggestive rather determinative, however.

5.4 Input Parameters

Our model is of the ARX variety (Autoregressive with Exogenous Inputs, discussed in Chapter 2), and we include two exogenous variables as well as variables derived from the price series itself which implicitly provide information about relative changes. Each input vector thus forms a distinct instance composed of multiple input factors. Our raw (pre-normalization) inputs are listed below:

1. S&P 500 Raw Index Value
2. 2-period percent change of S&P 500 Index
3. 5-period percent change of S&P 500 Index
4. Ratio of 2-period and 5-period Simple Moving Averages
5. Ratio of 5-period and 20-period Simple Moving Averages
6. Ratio of 7-period and 55-period Simple Moving Averages
7. Bar Summary: (high–low) / (close–open) * (volume / avg. volume)
8. 5-period percent change of Brent Crude Oil Closing Value
9. 5-period percent change of Trade Weighted U.S. Dollar Index

A dataset composed of inputs covering the date range from January 1992 through June of 1997 was used to test the value of various input parameters. Our starting list of candidate model inputs consisted of the fundamental and technical indicators used in [Brabazon06] along with several additional technical indicators we wished to investigate. While more sophisticated methods exist for determining the value of various model inputs, our methodology consisted of adding candidate inputs one at a time while using the 5-week

returns of the S&P 500 and the current index value as our base inputs. Input candidates which did not provide additional predictive value in terms of hit rate and RMSE were discarded. More information about the datasets used for our models is available in the appendix.

Several indicators which were expected to provide predictive value were dropped during preliminary testing after performing poorly according the above criteria. These include the CBOE Volatility Index (VIX), the Gold Fixing Price of the London Bullion Market and various interbank interest rates and bond yield spreads. The Bar Summary indicator is a customized indicator based primarily on intuition. Preliminary tests without it performed slightly worse than those where it was included. It was thus selected as an input.

Rather than recap the evidence supporting the influence of specific factors on market behavior, we will defer to previous research the justification for our input selection. See Brabazon and Kanas and Yannopoulos for some examples [Brabazon06, Kanas01]. We note that any choice of inputs, given the virtually infinite set from which to choose (or invent), will necessarily have a large subjective component.

5.5 Testing Dataset & Prediction Intervals

The dataset used to test our models covers the period from 1/15/1997 to 1/30/2015. The three primary historical datasets from which all indicators are derived are the S&P 500 index, Brent Crude Oil Futures, and the Trade Weighted US Dollar index for Major Currencies. The S&P 500 dataset contains attributes for the date, the weekly open, high, low and closing prices, and the weekly volume for each data row. The Brent and US Dollar datasets contain attributes for dates and closing prices only. The Brent and US Dollar datasets were obtained from the US Federal Reserve FRED database, while the S&P 500 dataset was obtained from Yahoo Finance. Each dataset is stored separately in an XML file and input arrays are built by deriving values from selected attributes at runtime. The correct (actual) values, which MLP outputs attempt to predict, are also determined at runtime via configuration parameters.

Lagged indicators of various lengths based on previous index values are used as inputs, and thus the date of the first prediction depends on the holdout data required by our slowest indicator (here, the longest moving average) and the starting date chosen. Thus, the starting point refers to the first vector stored when we run our program, and vectors utilized for training and testing occur later in the series after those required for indicator construction. Data are normalized prior to being input to the models.

Figure 5 provides snapshots of the XML files which are accessed at runtime, and the values derived from each are listed alongside the snapshots. Figure 6 shows the raw and

normalized versions of one input vector. Figure 7 provides a visualization of the MLP model.

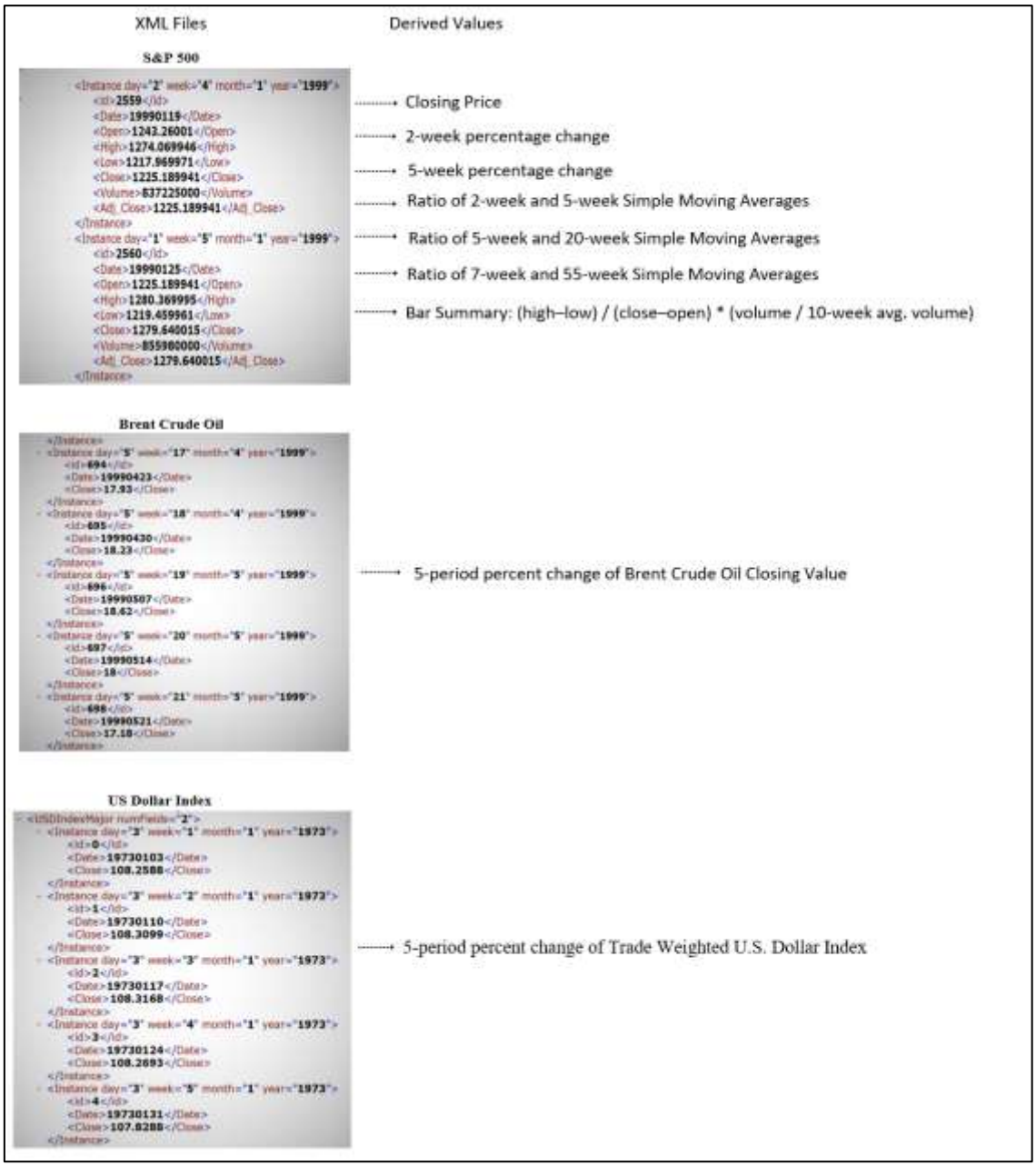

Figure 5: XML Files and Description of Values Derived from Each File

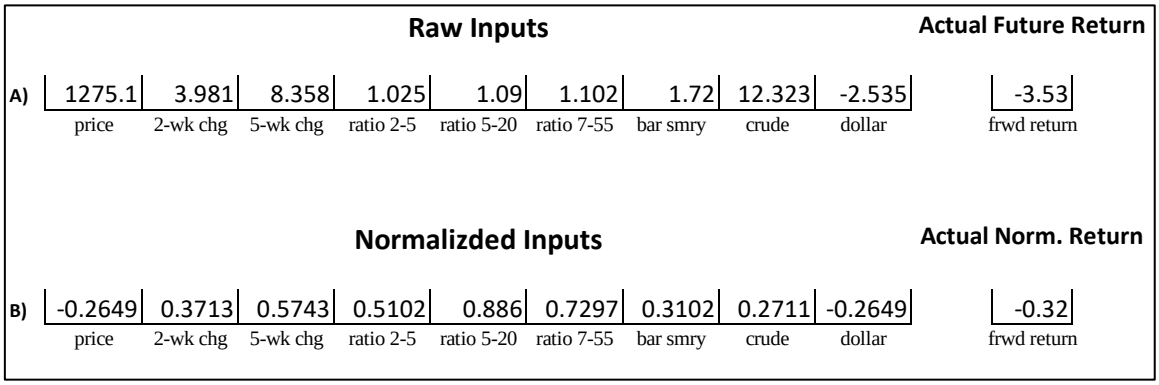

Figure 6: A) Raw Input Vectors B) Normalized Input Vector

Because we suspected performance is somewhat, perhaps largely, dependent upon our entry point (the point where we make our first prediction), we used several distinct starting dates.

Because our inter-market input attributes cross exchanges and international borders, missing values for some attributes occasionally occur on days for which the S&P Index trades due to differing holiday schedules of the various exchanges. We thus filter from the above period any instances for which all input attributes were not available, such that each input instance represents a trading day where all exchanges recorded values for the relevant indices. However, because we are using weekly data here, this process has minimal impact on the dataset.

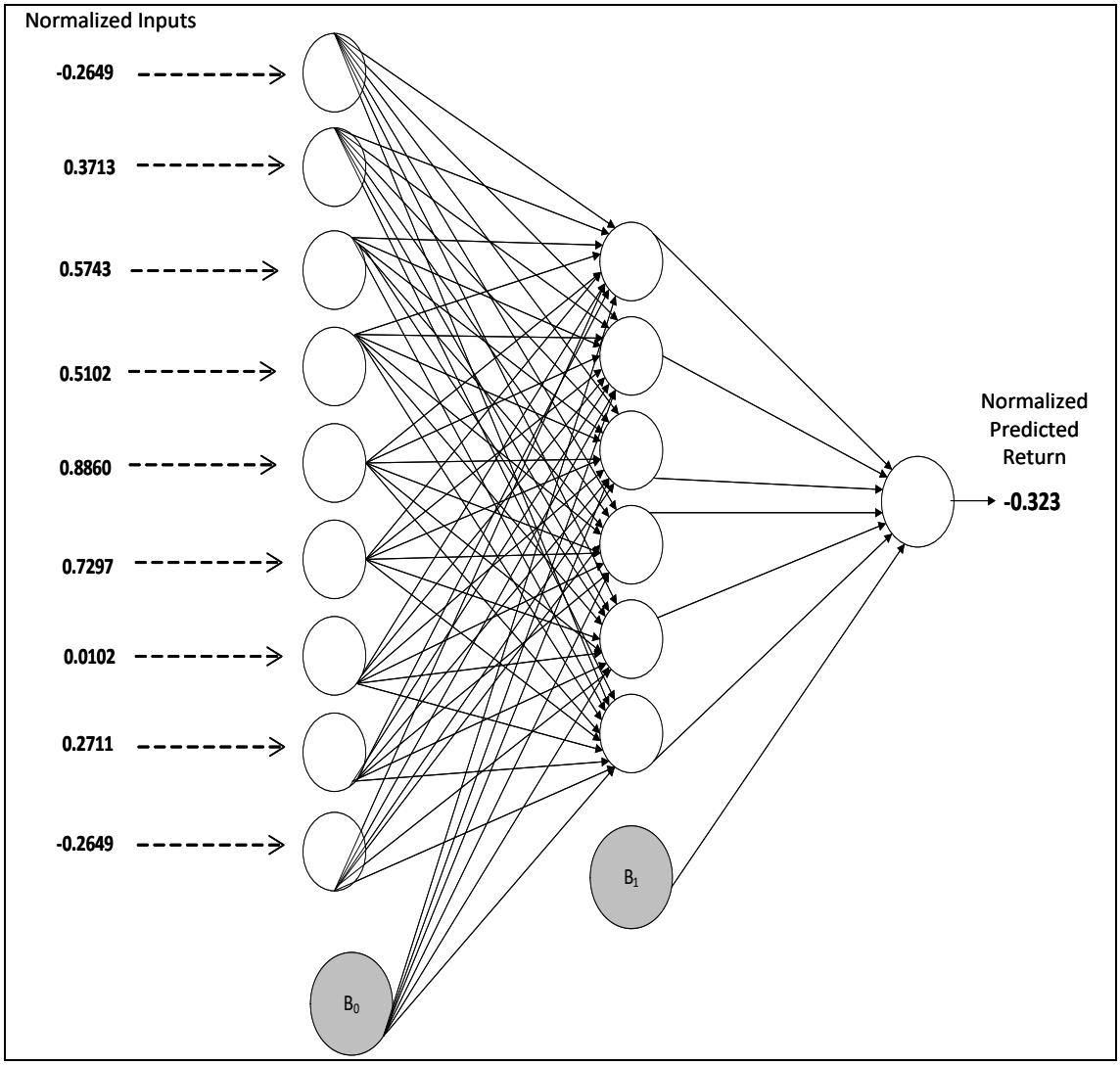

Figure 7: MLP Model. Inputs are from Figure 6

Unless otherwise stated, the periodicity of tests is weekly. Weeks were chosen rather than days because, while good results may be obtained over shorter periods with daily data, weekly data proved more conducive to accuracy over extended testing. Consequently, predictions and trading P&L calculations are made at five week intervals, which requires the trading system to stick with each prediction for a 5-week period.

5.6 Sliding Training and Testing Windows

In keeping with our strategic goal of long-term market outperformance, our MLP generates predictions every five weeks. Our design is uncommon to most research with ANNs related to market forecasting (though hardly unique in general) in that for each trained ensemble of MLPs a maximum of five predictions are made (a second round of testing makes only one prediction per trained ensemble). This is because we have chosen test windows of 25 calendar weeks, leaving us five 5-week prediction intervals within each test window. Thus, having trained the MLPs over a training window, we average the 5-week forward predictions of the MLP ensemble based on the first input vector from the test window, and we do this a total of five times (in sequence) within each test window, making trading decisions and P&L calculations after each prediction. We then role the training period forward 25-weeks, dropping off the first 25-weeks of data from our training window and incorporating the most recent 25 weeks. The MLP ensemble is then trained on this new training period from scratch, and new average predictions are provided for the next five 5-week intervals. And so on it goes for each test window.

No two runs over a complete out-of-sample test set are likely to produce the same returns. This is easy to see when you understand that predictions for every test window are the product of a freshly trained MLP ensemble initialized with non-fixed parameters (thanks, in part, to the use of RPROP as our training algorithm), and that there are many test windows within the entire out-of-sample test set. For this reason, the entire out-of-sample test set is traversed a total of thirty times (for our initial tests) in order to produce

a single test, with each run producing return, hit rate and other statistics. Our results for these statistics are reported as the averages of these thirty runs. We also report the best and worst run, in terms of returns, for each 30-run test. The expectation here is that, while every run may not produce returns above B&H when using a dynamic system with variably initialized parameters, we should expect the average run to outperform this benchmark if the selected model inputs maintain their predictive value over the full test set.

There are several advantages to the above design. Averaging MLP predictions has been shown to produce significantly better predictive performance than using a single predictor [Brabazon02], and using an average makes anomalous results much less likely. By always incorporating the most recent data, our MLP remains sensitive to underlying changes in market dynamics and should thus be more robust to time. Additionally, as every five predictions are the result of averaging the predictions of a uniquely trained ensemble of MLPs, the five predictions within any single test window can be said to be the result of a process fully independent from that of other test windows. The prospect of benefiting from an auspicious training run producing weight parameters which happen to be randomly well suited to the complete out-of-sample test set is largely foreclosed with this design.

5.7 Training Set Size

In chapter 2 we provided a metaphor for understanding market dynamics in the context of a biological example illustrating principles from Game Theory. We discussed the inability of any phenotype of side-blotched lizard to gain permanent reproductive advantage over other phenotypes and how the relative success of one phenotypic strategy portends its decline- and how the relative failure of another phenotypic strategy portends its success. Understanding that the metaphor is but a loose one, it is nevertheless useful when considering the length of our training period. As the underlying processes generating a price sequence become better understood by market participants and are subsequently altered as attempts to exploit this understanding increase, the understanding upon which these attempts at exploitation are based may cease, at least temporarily, to be valid. The resulting new situation is not likely to be novel, however, and we may expect at least its partial likeness to be revealed by the intermediate past.

If we accept this way of thinking about market dynamics, then there may be a danger in incorporating only a short history of examples in our training windows. This is because a short history will likely lead to an overweighting of recent relationships which may soon breakdown, and no information about relevant dynamics further historically removed will have been supplied to mitigate predictions predicated on these recent, but increasingly less valid, relationships. Of course, the more history we include, the further diluted currently valid short term relationships will become by more removed, perhaps less currently relevant examples. We may thus sacrifice short term precision by

attempting to sensitize our predictor to longer term dynamics. But as our strategic goal is to profit from major rather than minor trends, incorporating a longer history of training examples would seem to be a logical choice.

Preliminary testing with weekly data, however, does not support using a longer-term training history with DTT. Rather, this testing appears to support the idea that the dilution which occurs with extended training sets is more detrimental to accuracy/hit rate than are the informational limitations of shorter, more current training sets. While we can produce some runs which perform well using a longer training period, such results are few and far between. We get more consistent results, and thus presumably a more reliable predictor, by using a 25-week training period. The average run also appears to produce a higher final account balance, which is our ultimate measurement when attempting to challenge EMH.

Dynamic Validation somewhat mitigates the issue of historical information loss caused by shorter training sets. This is because the predictor (ensemble) is selected from amongst 5 MLP ensembles which, while each is of a length of only 25 weeks, are taken from a set of 125 weeks. In essence, Dynamic validation allows us to cherry-pick the more relevant short-term history from amongst a larger history based on its performance on the validation set. Of course, should the validation set be distinctly different in character from the test set, then we would expect our selected ensemble to perform poorly on that test.

5.8 Training Algorithm

Because we are training an ensemble of MLPs on many overlapping datasets, our methodology benefits from a training algorithm untethered to a single set of training parameters. If we were to use a standard backpropagation algorithm, any particular settings for learning rate and momentum may, or may not, train any one MLP to perform on the training set with sufficiently minimal error to generalize to a test set. While we introduce early stop strategies for MLPs failing to train to our predetermined error rate, putting such insufficiently trained MLPs into service is a means of last resort. As discussed in chapter 2, resilient backpropagation, which does not require us to specify rates for learning rate or momentum, provides us with the best option for training a large number of MLPs for each of many training windows without predetermining appropriate parameter values for learning rate and momentum in advance.

As mentioned above, a consequence of using RPROP as our training algorithm is that no two MLPs, let alone MLP ensembles, will necessarily be weighted identically on successive runs over a test set. If our MLP ensembles were constructed with fixed initialization parameters for each structure within the ensemble, then we could expect repeated runs over identical datasets to produce identical results. This, indeed, is how most MLP models are built. This is logical, particularly where the properties of the dataset are expected to be relatively stable. Market returns do not likely result from a stable data generation process, however, and we have little confidence that parameters optimized for our test dataset would be optimal in the future. There may thus be merit in

repeating performance over the same dataset with MLPs whose initialization parameters are varied with each test window and with each test run. This requires each test be composed of multiple runs over the test dataset, however, and leaves us with a range of results rather than a single result for each test. Performance measures are thus averaged over all runs for a given test over a given test period, and highs and lows are reported for return results.

5.9 Hidden Layers & Hidden Nodes

Previous research and preliminary testing suggest we use no more than one hidden layer (HL) in our MLP structures. While, in some cases, we found a second hidden layer with a large number of nodes benefited DTT during preliminary testing, these results were not consistent enough to warrant inclusion in our study. Our model contains a single hidden layer of six hidden nodes.

The model was arrived at by starting with the heuristic of taking the mean of the input and output layers suggested in [Heaton08] and through preliminary testing over the dataset used during input selection. The six hidden nodes also comport with [Brabazon06] where the structure used consists of 10 inputs and the output, as with our MLPs, is continuous. We have therefore borrowed, albeit after much preliminary testing, our structure from [Brabazon06], minus one input node. Our input factors, however, are distinct.

While it can often make sense to prune noncontributing connections from an MLP structure, that is not the case here. The number of MLPs required by our design, the potentially changing dynamics underlying market price discovery combined with the use of a fixed MLP structure, and the specification dictating that only a few predictions are made per uniquely trained MLP ensemble makes pruning both impractical and undesirable. Thus, all of our MLPs are fully connected.

5.10 Output Layer

MLPs produce continuous output for this model, and each singular output from each MLP is taken as a prediction of the index's 5-week forward return. The outputs from all MLPs in an ensemble (one ensemble for every training window) are then averaged to produce the final prediction used to make long or short trading decisions. By providing a measure of magnitude to return predictions, rather than merely a binary choice between positive or negative outcomes, continuous output allows us to apply thresholding to our trading decisions. Additionally, customized features of our methodology pertaining to data normalization and dynamically constructed datasets integrate poorly with Encog's classification features.

5.11 Data Normalization and Activation Function

Data normalization most often squeezes input values into a range of [0, 1] or [-1, 1], and it is common to use a sigmoidal function for the former range and the hyperbolic tangent (TANH) function for the latter. Because we are predicting both positive and negative returns, TANH is a natural choice. However, as normalized values are de-normalized prior to signal processing, there is no mathematical reason to insist on the wider range.

A fact about data normalization of which we were unaware, but that became clear during preliminary testing, is that normalization is best done in segments when the dataset extends over many years and the range of values expands over time- rather than normalizing all input vectors over the entire dataset. It should be obvious that it makes little sense to train an MLP that will make predictions on S&P 500 index values from, say, 1999 with data vectors normalized using high values which include data from the year 2016, or with low values that include the year 1950. Such values are simply not within the range of possible outcomes we might reasonably expect the market to produce in the year 1999. The issue is particularly pronounced where we use raw rather than percentage change values as inputs, but it is also possible that percentage change fluctuations behave differently as the range of raw historical values increases over time. For this reason, we normalize vector inputs prior to training each MLP ensemble based on the data values from the period upon which each ensemble will be trained.

5.12 Trading System and P&L Calculations

The S&P 500 tracking security we trade in order to calculate returns is an Exchange Traded Fund that goes by the symbol SPY. This ETF trades at one tenth the value of the S&P 500 futures contract and thus allows us to begin trading with a modest trading account balance of $10,000. We do not take any direct account of dividend distributions or costs related to short selling in our calculations. Trading costs are computed to be $10.00 per trade such that a completed trade, entry and exit, amounts to a $20.00 reduction of our trading account balance from which we purchase shares or sell them short. Unlike a buy & hold market strategy where the number of shares would remain constant regardless of their price (assuming no dividend reinvestment), a long-short strategy creates fluctuations in the number of shares traded over time. A successful short trade increases our account balance at the same time the share price is falling, allowing us to buy more shares once we reverse our position. A failed short trade, where the price continues to rise against our short position, reduces our purchasing power when we reverse our market position. We thus must determine the number of shares we can buy or sell short, given our current account balance, after exiting each trade and prior to entering a new one.

Our strategy is always ‘in-the-market’, meaning that at no time will we be sitting in cash. Rather, MLP ensemble signals above zero cause us to either continue or reverse into a long market position, and signals below zero cause us to either continue or reverse into a short position. These signals, however, are only provided at the end of each prediction

period. Thus, when we get a buy signal (MLP output > 0), we maintain a long position for a minimum of 5 weeks (as we are predicting returns 5 weeks forward), and a new signal is not generated until the end of these 5 weeks. At that time, we decide to either stay in our current position or reverse to the opposite position depending on the sign of the new signal.

Chapter 6

RESULTS AND ANALYSIS

Before discussing our results, it may be helpful to visualize the composition of an averaged equity curve in order to make clear what each ending equity balance represents. Figure 8 displays thirty equity curves composing the average curve of a single test for which the average ending equity balance (represented by the thick blue line) is reported.

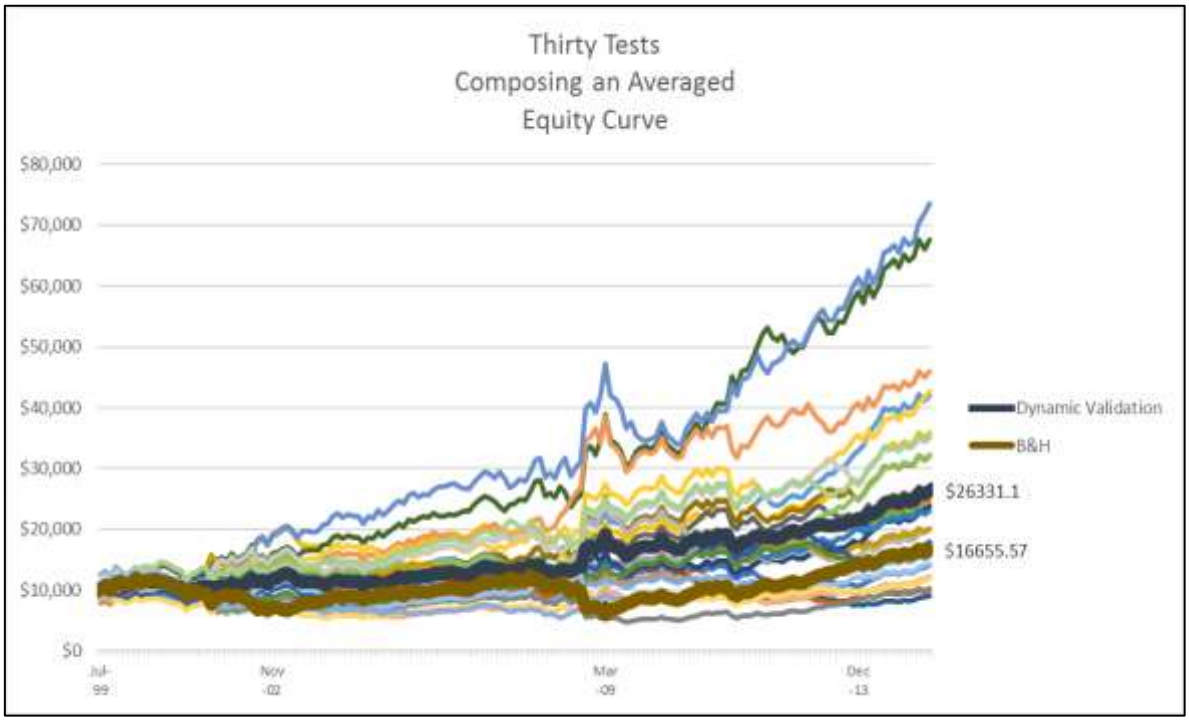

Figure 8: Average Equity Curve Composition: Thirty Runs Compose a Single Test

Figure 8 demonstrates one way in which our results must be distinguished from other studies. Because training is dynamic and RPROP does not fix parameters for learning

rate and momentum, no two runs are likely produce the same sequence of predictions. We must, therefore, look at the equity curves from multiple runs to see how our predictor would have likely performed over a given period. While a range of equity curves (or their average) is perhaps less satisfying than a single result, we believe these provide a more realistic expectation for how a real-time system might perform.

6.1 Dynamic Training & Testing

Our tests for DTT cover several date ranges in order to assess the robustness of our results to the varied features of different test sets. The importance of this approach can be seen by looking at the DTT and B&H equity curves produced by a preliminary test which covers January 1999 through February 2009, and comparing this with other results achieved using DTT.

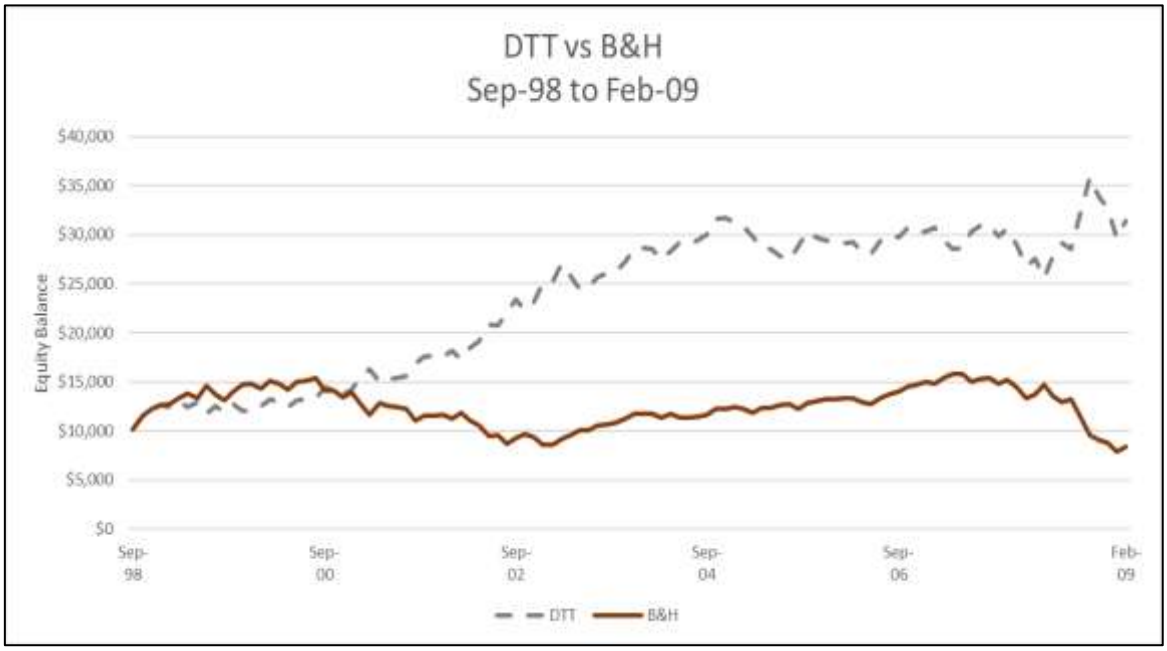

Figure 9: DTT Avg. Equity Curve vs. B&H, Jan-99 to Feb-09

In figure 9 we see DTT produces, on average, returns well above those produced by a B&H strategy for the period. Our chart stops right near the bottom of the 2007 – 2009 market downturn (and crash), where B&H returns from January 1999 were approximately -37%. In contrast, our long-short strategy produces returns of just over 59%, and these appear to be on the rise as this chart ends.

This would be a good place to declare victory and go to press. But if we extend our test out through December of 2015 we see that our superior performance, while extending over multiple years, is nonetheless transitory.

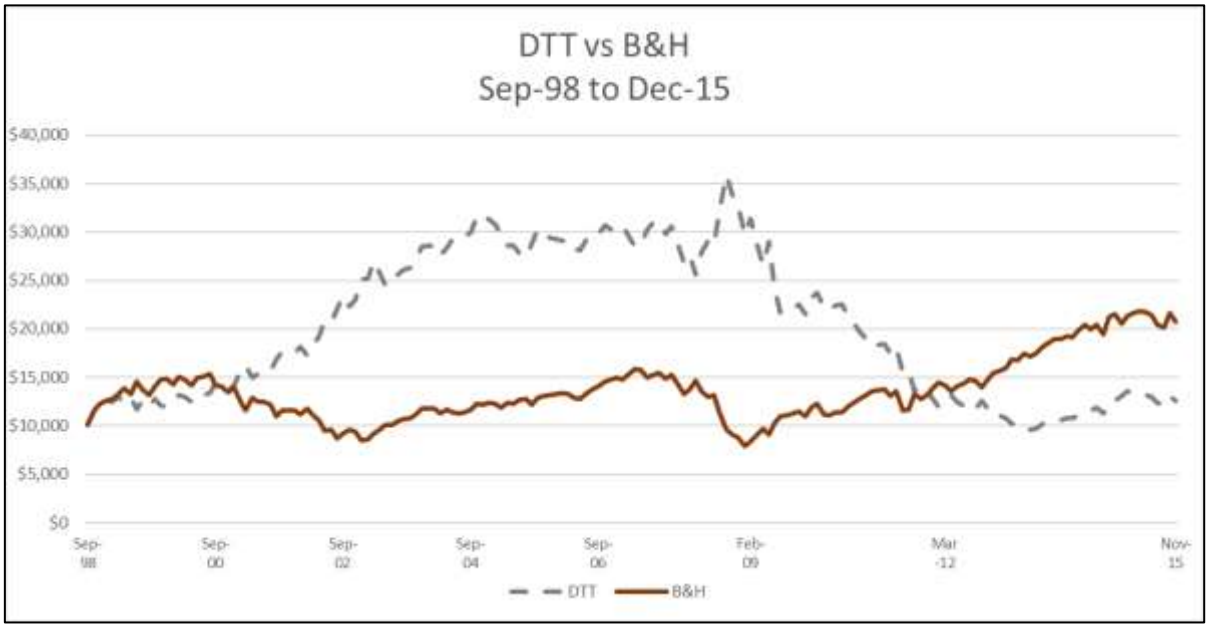

Figure 10: Averaged Equity Curve, 30 runs

We see in Figure 10 that while DTT maintains above market returns over most of this 16-year period, these returns level off in late 2004 and fall below those of a booming market in early 2012. As importantly, returns for the long-short strategy from 2004 forward are net negative. Thus, we turn from good performance to breakeven performance after 2005, and then to outright poor performance after early 2009. Put another way, the same structure with the same inputs, while being repeatedly retrained over the period, ceases to produce reliable predictions seven years into the 17-year test period.

It is important to remember that each of the above figures shows the average of thirty equity curves, and are thus unlikely to be anomalous. Indeed, the shift in fortunes of

DTT around the mid-2000s may suggest a change in the underlying data generation processes by which index prices occur.

As we vary the start dates of our test runs the limitations of this predictive system become clear. Table 2 summarizes the full list of tests performed with DTT. Despite extraordinary returns for a run starting in 1997 with the single layer structure, the remainder of the table demonstrates that such returns could not have been achieved reliably with this system. Seeing that such returns are nonetheless achievable and repeatable for select start dates, despite being produced by an otherwise unreliable system, we might reasonably question the practice of reporting a system's returns for a single test over a fixed testing period.

Dynamic Training & Testing											
Start Date	End Date	Avg. Hit Rate	Avg. Hit Pct	Max ending Equity	Min Ending Equity	Avg. DTT Ending Equity	end_B&H equity	Total Rolling Return	Total B&H Return	# Runs Beat B&H	DTT vs B&H
Jan-97	Mar-15	1.39	0.58	$61,151	$47,288	$47,224	$27,617	372.2%	176.2%	29/30	$19,607
Jan-98	Apr-15	1.07	0.52	$12,611	$3,896	$8,331	$21,652	-16.7%	116.5%	0/30	($13,321)
Sep-98	Dec-15	1.18	0.54	$16,872	$10,457	$12,544	$20,728	25.4%	107.3%	0/30	($8,184)
Jan-99	May-15	1.34	0.57	$28,614	$11,436	$19,990	$17,406	99.9%	74.1%	24/30	$2,584
Jun-04	Jun-15	1.03	0.51	$5,734	$2,641	$3,867	$18,847	-61.3%	88.5%	0/30	($14,980)

Table 2: Dynamic Training & Testing

6.2 Dynamic Validation

For our tests employing Dynamic Validation, S&P 500 closing values are removed as inputs, because the complete training set takes place over a two-and-a-half-year period.

Price values earlier in the training sets thus tend to be far removed from those of the test sets and will not reflect realistic prices for a test set that occurs significantly later in the date range. As with DTT, Dynamic Validation testing was done using multiple start dates.

6.2.1 Results with Dynamic Validation

Dynamic Validation appears to do a much better job of beating B&H, excluding costs from holding short positions on ex-dividend dates, for starting dates between 1997 and 2001. This performance breaks down, however, where our tests begin after 2001. Table 3 provides detailed averages for the thirty runs performed for each test interval.

Test #	Start Date	End Date	Avg. Hit Rate	Avg. Hit Pct	Max DV End Equity	Min DV End Equity	DV Avg. Ending Equity	End B&H Equity	Total DV Return	Total B&H Return	# Runs Beat B&H	DV vs B&H
1	Jan-97	Mar-15	1.25	0.55	$69,557	$11,096	$36,559	$27,617	265.6%	176.2%	21/30	$8,941
2	Jan-98	Apr-15	1.21	0.55	$44,608	$9,793	$21,805	$21,652	118.1%	116.5%	13/30	$154
3	Jan-99	May-15	1.41	0.58	$53,979	$15,240	$26,928	$16,656	169.3%	66.6%	27/30	$10,273
4	Jan-00	May-15	1.2	0.54	$30,919	$4,970	$17,937	$14,859	79.4%	48.6%	22/30	$3,078
5	Jan-01	May-15	1.46	0.59	$44,050	$16,494	$27,520	$16,438	175.2%	64.4%	30/30	$11,082
6	Jan-02	Jun-15	1.24	0.55	$25,393	$7,047	$13,636	$18,779	36.4%	87.8%	4/30	($5,143)
7	Dec-03	Dec-15	1.35	0.57	$32,256	$4,293	$15,041	$18,514	50.4%	85.1%	5/30	($3,473)
8	Mar-04	Apr-15	1.23	0.55	$19,139	$3,968	$10,949	$18,809	9.5%	88.1%	1/30	($7,860)
9	Sep-04	Apr-15	1.34	0.57	$31,712	$6,569	$16,212	$19,129	62.1%	91.3%	6/30	($2,917)
10	Jun-05	Jan-16	1.12	0.53	$14,771	$4,757	$10,104	$16,964	1.0%	69.6%	0/30	($6,860)

Table 3: Dynamic Validation Test Results. The neural model for DV Tests is 9-6-1

While model output is continuous, hit rate essentially evaluates the model's success as a binary classifier and is equivalent to accuracy. Looking at the above table, we see that hit

rates remain well above 1.0 for all tests, and that hit percentages (percentage of directional moves predicted correctly) are between 54% and 59% irrespective of our average ending equity being above or below that of B&H. The differences in average ending equity balances, and thus performance- while being correlated with hit rates at a 0.596 PPMCC- are best predicted by the ratios of outperforming tests to total tests. As is to be expected, these are greater where our average ending equity is higher, and they suggest our model, despite its lack of fixed initialization parameters, has some measure of reliability over these intervals. Clearly, if we wish to employ a similar model in live trading, or should we wish to challenge the notion of an efficient market, we must understand what distinguishes these intervals from those where our model performs poorly.

6.3 Analysis of Dynamic Validation

What is most striking about the results from Dynamic Validation shown in Table 3 is not the difference in outcomes between runs starting earlier from those begun later, but rather that the latter intervals, despite apparently being mere subsets of the former, should be those for which the model performs poorly. After all, Dynamic Validation seems to have performed quite well over these very same timeframes where testing started prior to 2002, so why should it perform so poorly when tests are begun in 2003 and beyond?

To see just how dramatic this difference in performance is over the post-2001 period relative to tests begun earlier, we can transpose the averaged equity curves for tests 5 and

6 from table 3, as shown in Figure 11. What this shows is that, despite maintaining comparable performance through 2006, the equity curve from test 6 begins underperforming around this time. The reason for this difference points to a unique challenge with testing predictive systems' ability to produce excess returns.

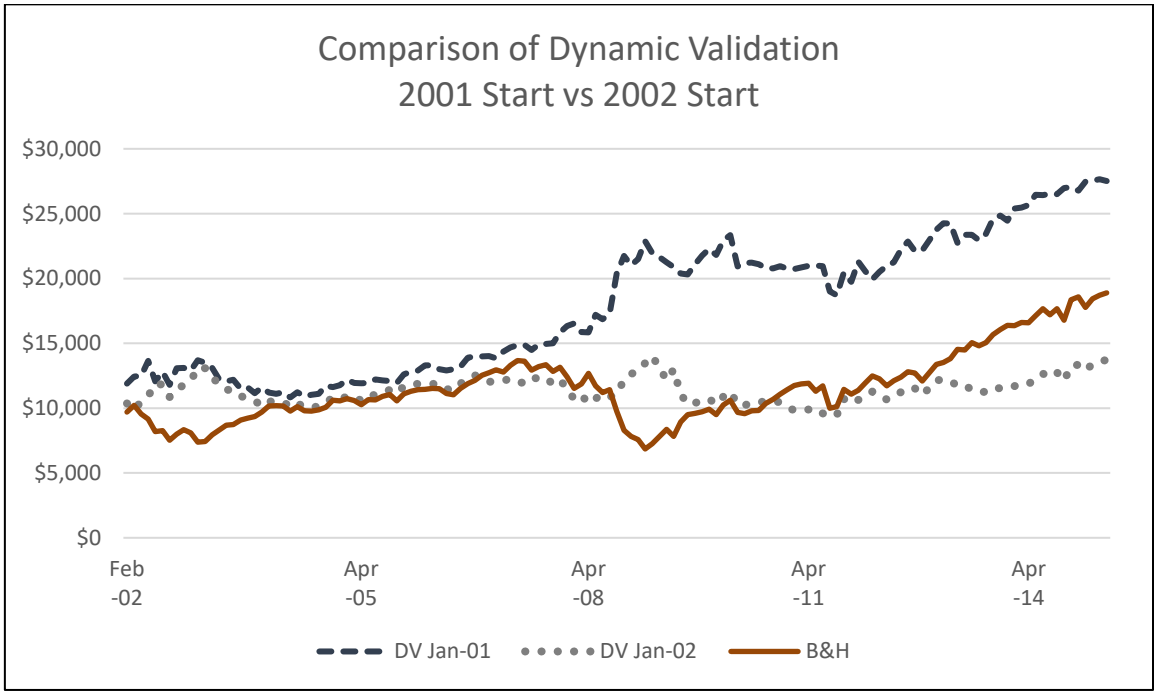

Figure 11: Comparison of Two runs of DV Offset by One Year

Remember that each MLP ensemble makes, in our tests here, a total of five predictions over a 25-week period prior to being retrained for further predictions. While it is true that, from January 2002 forward, the MLP ensemble producing the equity curve which begins in 2002 spans the same time interval as the curve which began in 2001, the inputs and outputs, as well as the training and validation datasets, are not identical because the dates from which the inputs are taken are offset from between one to four weeks. Thus, the holdout, training and validation periods encompass slightly different dates for the

MLP ensembles producing the two curves, and the input/output pairs of the test datasets do not correspond. For example, whereas 2/08/2002 is the first output date in the series of five predictions for curve 5, it is the third in the series for curve 6. This means the MLP ensemble for curve 5 is validated against the period from 7/6/2001 through 1/4/2004 for the 25-week test set encompassing 2/8/2002, whereas the ensemble for curve 6 is validated against 4/27/2001 through 10/26/2001 for the test set encompassing that date. This makes for a 10-week difference, not only between validation sets, but for the training sets and the hold-out data used to produce the indicator inputs to our MLP ensembles. This difference in data history persists over the entire test period through 2015.

We can demonstrate that it is indeed this 10-week difference that produces the disparity in our average returns by running a fresh set of 30 runs after aligning the historical data of curve 6 with that of curve 5. We can see in Figure 12 that this alignment indeed produces forward returns similar to curve 5.

One rationale for testing multiple start dates was the concern that return results may suffer if a system is initially deployed during a market period for which it is poorly suited. We can say, here, that our variability in performance is related to slight variability in the datasets of each overlapping date range, rather than to any particular date of deployment.

The reader may be tempted to conclude that the dependency of our results on the historical date ranges used by our system is merely a flaw in methodology. After all, there is no requirement that five predictions be made for each validated MLP ensemble. Couldn't the problem be solved by simply by making one prediction for each ensemble, thus eliminating the vicissitudes of returns occasioned by variability in our data histories? Partly, yes.

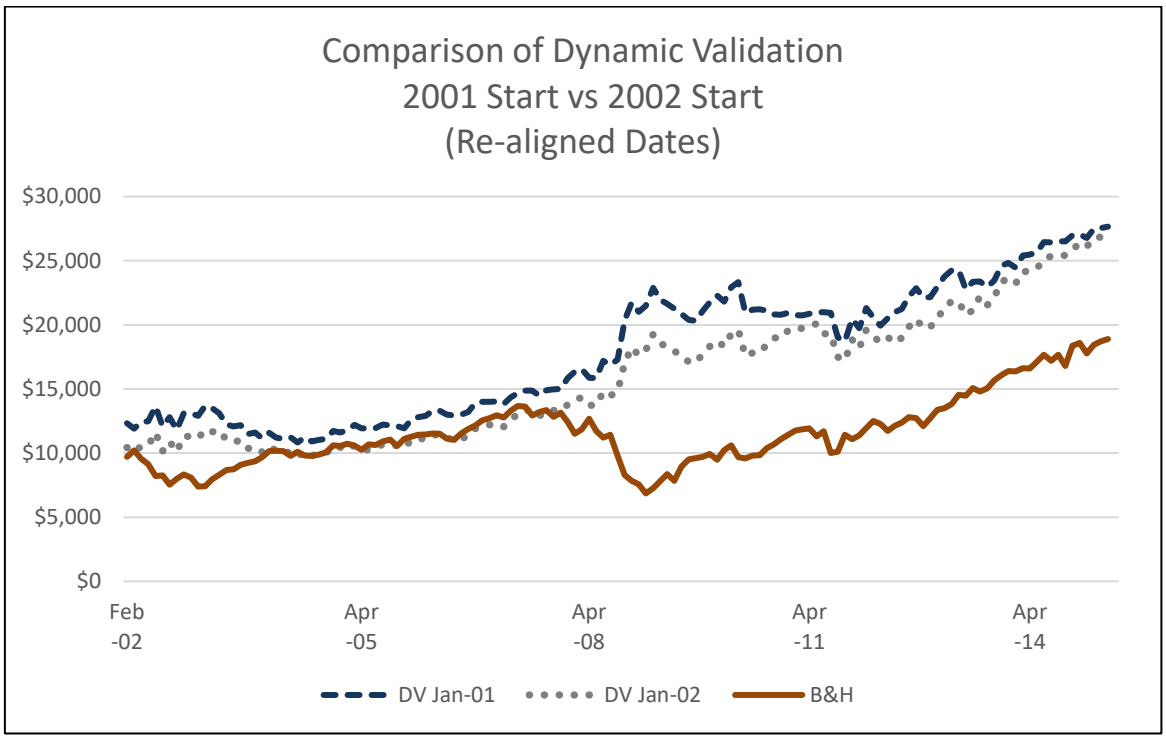

Figure 12: Re-aligned Input Dates

Modifying DV such that only one prediction is made per trained MLP ensemble, we get results comparable to those of Figure 12. Yet, the 5-week forward prediction still creates five sets of mutually exclusive input/output pairs, and a test can only cover one set of pairings over a single run. For example, if the first input date in 2002 is 2/8, both the

output and next input date will be 3/22, and the output for 3/22 will occur on 4/26. The input/output pairings occurring in-between these dates (2/22 and 3/29, for example) will not have been part of the test. This issue of distinct pairings exists only for testing datasets, where MLP output result in trading decisions and where P&L calculations must be made. For training and validation purposes, all pairings are used.

Despite this issue, the differences in datasets are significantly less when we make only one prediction per trained ensemble. Moreover, as there are only five possible datasets for a given date range for 5-week forward predictions, we can test each of these. We do so with our third set of tests. However, with an eye toward improving return performance, we make a slight modification to our trading methodology.

6.4 Single Prediction Dynamic Validation with Thresholding

As we noted in chapter 2, previous research indicates that ANN forecasts may perform relatively poorly in markets trending strongly upward. When we consider returns, rather than just error rates, this problem may be exasperated. In particular, if our market strategy involves shorting or exiting the market based on the predictive signals produced by our ANN model, then signals incorrectly predicting negative returns are costly in upward trending markets, where they are less likely to be correct. We see in Figure 12 that the returns with DV, despite being above buy-and-hold, are mostly flat from 2003 through 2007 when the market was in a boom period. Thus, despite having moved ahead of buy-and-hold during the previous downturn, DV fails to benefit from the upward

trend. We find this repeatedly when we breakdown individual tests. It is clear that the model produces signals which, while often correct, too often predict negative returns during upward trending markets.

One method of mitigating the impact of negative predictions in upward markets is to insist such predictions cross a certain magnitude threshold before they are considered actionable. A threshold merely changes the rules governing how signals are acted upon (in terms of trading), rather than the production of the signals themselves. It has no bearing, as used here, on the neural model. As part of a system, ANN based or otherwise, attempting to overcome the constraints of EMH, however, it is an acceptable tool.

If you remember, our strategic goal is to obtain at or near market returns in upward markets while also profiting from downward moves. Large downward moves- while often dramatic in magnitude- tend to be more short-lived than upward trends. By thresholding negative MLP signals such that they are not considered actionable short of a specified magnitude, we may be able to minimize the impact of incorrect negative signals during upward trends. The obvious cost of this technique is that the system is slower to act upon accurate negative signals, and thus may be slower to respond to downturns, or perhaps even miss them entirely.

We have also discussed a concept we coined “Rules Bias” in previous chapters. A problem here, if we wish to add a threshold to the system, is that we have already gotten

a partial view of the system's performance over the complete test set. How can we now implement a threshold without committing the error of back-fitting our rules to suit our predictor's behavior?

Remember in Chapter 5 we set our threshold in advance of running our tests. This eliminates any potential bias produced by observing our predictor's behavior on the test set, so long as we do not then vary our threshold to improve returns. Here, we do not.

As discussed in the last chapter, for Single Prediction DV we chose a -1.0% threshold for acting upon negative signals. Consequently, should our MLP ensemble predict a -0.8% return for any 5-week forward period, for example, this signal would not be strong enough to induce our trading system to exit a long market position and go short. If we were already in a short position, then this -0.8% signal would cause us to reverse into a long position. We should note that, while our trading behavior is in this way different, our MLP measurements are unchanged. A -0.8% MLP signal, while not inducing a short position, is still considered a hit if the 5-week forward returns are at all negative, and it is a miss if they are positive. Hence, our hit rates and hit percentage calculations still reflect the accuracy of the MLP ensemble's sign predictions. Only the P&L calculations are different.

Single Prediction DV takes considerably more time and resources per run. For this reason, we limit our ensemble size to twelve and perform only 10 runs per test, rather than 30. We perform tests for four different years (1999, 2001, 2003, and 2005).

Anticipating that our MLPs may be sensitive to each of the five datasets available within a given date range, we perform five tests over five consecutive weeks for each of these years in order that every test dataset (or set of input/output pairings) is tested for each starting year. Table 4 shows the results of these tests.

Start Date	End Date	Avg. Hit Rate	Avg. Hit Pct	Max DV End Equity	Min DV End Equity	DV Avg. Ending Equity	End B&H Equity	Total Avg DV Return	Total B&H Return	# Runs Beat B&H	DV vs B&H
1999											
2/26/99	12/11/15	1.3	0.56	$41,106	$15,008	$25,626	$16,300	156.3%	63.0%	7/10	$9,326
3/5/99	11/13/15	1.18	0.54	$27,031	$9,884	$18,593	$15,982	85.9%	59.8%	7/10	$2,611
3/12/99	10/16/15	1.15	0.53	$31,886	$9,128	$21,885	$15,858	118.9%	58.6%	9/10	$6,027
3/19/99	9/18/15	1.46	0.59	$31,670	$84,058	$53,747	$15,077	437.5%	50.8%	10/10	$38,670
3/26/99	8/21/15	1.45	0.59	$47,831	$20,315	$27,630	$15,373	176.3%	53.7%	10/10	$12,257
2001											
2/23/01	1/8/16	1.54	0.61	$45,759	$11,357	$28,827	$15,568	188.3%	55.7%	9/10	$13,259
3/2/01	12/11/15	1.31	0.57	$28,809	$12,891	$19,523	$16,502	95.2%	65.0%	9/10	$3,021
3/9/01	11/13/15	1.22	0.55	$25,539	$7,985	$17,708	$16,589	77.1%	65.9%	5/10	$1,119
3/16/01	10/16/15	1.18	0.54	$37,684	$11,373	$18,580	$17,688	85.8%	76.9%	4/10	$892
3/24/01	9/18/15	1.41	0.58	$46,776	$17,833	$33,310	$17,230	233.1%	72.3%	10/10	$16,080
2003											
2/21/03	1/29/16	1.24	0.55	$36,398	$10,187	$21,805	$22,894	118.1%	128.9%	3/10	($1,089)
2/28/03	1/1/16	1.42	0.59	$39,297	$18,869	$26,316	$24,322	163.2%	143.2%	5/10	$1,994
3/7/03	12/4/15	1.29	0.56	$26,450	$10,077	$16,721	$25,310	67.2%	153.1%	1/10	($8,589)
3/14/03	11/6/15	1.21	0.55	$23,794	$11,217	$18,265	$25,400	82.7%	154.0%	0/10	($7,135)
3/21/03	10/9/15	1.24	0.55	$23,173	$8,971	$15,224	$22,567	52.2%	125.7%	2/10	($7,343)
2005											
2/18/05	2/26/16	1.34	0.57	$9,963	$23,190	$15,730	$16,363	57.3%	63.6%	5/5	($633)
2/25/05	1/29/16	1.26	0.56	$26,602	$12,693	$18,655	$16,104	86.6%	61.0%	8/10	$2,551
3/4/05	1/1/16	1.57	0.61	$33,603	$20,335	$26,306	$16,760	163.1%	67.6%	10/10	$9,546
3/11/05	1/8/16	1.38	0.58	$28,139	$9,222	$18,527	$16,145	85.3%	61.5%	7/10	$2,382
3/1/8/05	1/15/16	1.27	0.56	$26,595	$14,688	$19,054	$15,983	90.5%	59.8%	8/10	$3,071

Table 4: Single Prediction DV with Thresholding

Single Prediction DV with Thresholding appears to produce better results than the DTT method, but it underperforms B&H for four of the five subsets in 2003. For each year except 1999, at least one test performs worse or only marginally better than B&H. For every year, the results vary substantially depending on the start week chosen.

6.4.1 Analysis of Single Prediction Dynamic Validation with Threshold

Single Prediction DV minimizes the differences in training sets, validation sets and indicator inputs such that they only differ by a single 5-week period from one input date to the next. This being the case, it would be surprising to get such divergent prediction results between input weeks simply as a result of such minor differences in historical datasets. The alternative explanation is that the differences in test datasets (the five input/output pair sequences resulting from 5-week forward predictions) are the source of the high variability in performance. Figure 13 provides a visualization of how the datasets for the 2003 tests differ for the first three predictions of each sequence.

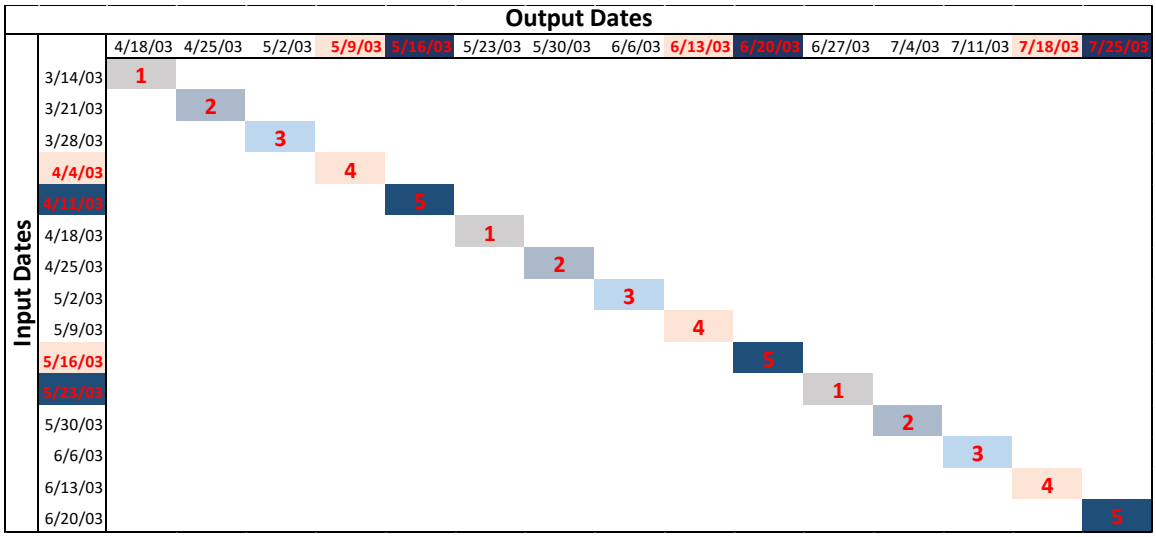

Figure 13: Five Distinct Test Sets Resulting from 5-week Forward Predictions

Looking at average equity curves for 2003 in table 4, we see a large difference between the average equity curves starting on February 28 and on March 7. If we take the best run (of the ten runs composing the average) from the February 28th curve and the worst

run from the March 7th curve, we can compare how they differ. For example, if the divergence in return performance between prediction sequences, offset by a mere week, is the result of variability in the historical dataset, then we would expect to see a large number of differences between the sign predictions made by our MLP ensembles for each sequence. On the other hand, if the divergence is related more to the variability of input/output pairings between the two test datasets, then we would expect there would be poor correspondence between the outcomes of those sign predictions, as measured by the number of hits and misses that correspond to one another between sequences.

For the 132 predictions made for each of these datasets, the predicted market directions from a given week of the first dataset (starting 2/28/03) are equal to the predictions made the very next week (starting 3/7/03) a total of 107 times. However, the outcomes of these predictions (i.e. whether they are hits or misses) are equivalent between these datasets only 83 times out of the 134 predictions. Thus, the more impactful factor accounting for the divergence in return performance between these two prediction sequences is the difference between input/output pairs (the subsets of the complete test set), despite these being offset by a mere week.

Start Date	Hits	Misses	# equal predictions	# equal outcomes
2/28/2003	76	56	107	83
3/7/2003	75	57		

Table 5: Best Run 2/28/03 vs Worst Run 3/7/03

It appears, then, that our MLP structure is only effective at making predictions which produce above market returns for a subset (or several subsets) of the weekly market data covering the date ranges for which we have conducted tests, and success is dependent on when during the date range the system is deployed. In conducting our preliminary tests to find a suitable structure, date ranges were typically incremented by multiples of 5 (25, 50, 100), and this practice created many test sets composed of the same input/output pairs where the date ranges overlapped between tests. And while our structure seems to work effectively in many cases, it is ineffective for others.

6.5 Revisiting DTT (Analysis)

Results which indicate a high variability in return performance for Dynamic Validation (and for Single Prediction Dynamic Validation with Thresholding) raise a question: Does this performance variability hold with other ANN testing methodologies? Does it hold for DTT? Recall that, despite one outstanding average equity return curve from 1997 through 2015, most tests starting later in the testing date range underperformed B&H for DTT tests. Is the over performing test using a different set of input/output pairs than the others, and would our performance change if we were to align these pairings for the later tests?

Rerunning each of the DTT tests after aligning both the input/output dates and the historical datasets (because we are back to using MLP ensembles which make five predictions each before retraining) with those of the 1997 test provides an answer (here,

we use tests of 10 runs each). As can be seen in Table 6, DTT performs quite well for all date ranges when the test and historical datasets are aligned with those of the 1997 test.

DTT Results (Dates Aligned with 1997 Test)											
Start Date	End Date	Avg. Hit Rate	Avg. Hit Pct	Max. Ending Equity	Min Ending Equity	Avg. Ending Equity	End B&H Equity	Total Rolling Return	Total B&H Return	# Runs Beat B&H	Rolling vs B&H
Jan-98	May-14	1.33	0.57	$36,212	$11,720	$22,109	$19,718	121.1%	97.2%	5/10	$2,391
Sep-98	Dec-14	1.09	0.52	$32,084	$12,581	$21,718	$19,386	117.2%	93.9%	6/10	$2,332
Jan-99	May-15	1.33	0.57	$37,078	$13,441	$26,007	$16,656	160.1%	66.6%	9/10	$9,351
Mar-04	Sep-15	1.27	0.56	$39,600	$15,999	$23,925	$17,818	139.3%	78.2%	8/10	$6,107

Table 6: Re-tests with DTT

What appeared to be an extremely poor predictive ANN model appears on second look to have similar predictive ability to that of Dynamic Validation for certain input/output sequences, and we are able to obtain good results without the use of thresholding. However, the limitation inherent in both our models; namely, that their performance is highly dependent on which of the five possible test sets is chosen within a given date range as well as on when the tests are begun, most certainly dooms any hopes of either model generalizing reliably to live markets in their current form. Yet it is this very limitation that may provide us with our most valuable insight into market forecasting with ANN models.

6.6 Discussion

It is important to understand that the results produced here do not account for the costs of being short the SPY (the S&P 500 tracking ETF used as the trading vehicle) on ex-dividend dates. Not only does being short a security on ex-dividend dates cause the short seller to lose the dividend (along with compounding benefits from reinvesting it), being short on ex-dividend dates also requires that the holder of the short position pay the dividend. Obviously, this complicates hypothetical return calculations over a long period. The important thing to take away, however, is that simply beating B&H by some marginal dollar amount may not, in fact, beat B&H after these costs are considered. Any trading performance successful enough to challenge EMH would be required to account for these costs. As the results produced here do not warrant such a claim, we ignore these costs for the remainder of the discussion.

The problem of having more than one test set within a date range is purely a function of making predictions more than one period forward. Predictions with weekly data made on a weekly basis with accompanying trading decisions and return calculations can ever only be made using a single test set for a specified date range. For shorter timeframes of one to two years, finding a structure which produces reliably good 1-week forward predictions proved achievable in preliminary testing. However, we chose 5-week forward predictions because we were unable to obtain reliable returns over the extended date range from which our test sets are drawn using single week predictions. By coincidentally dividing this dataset into fifths, we also made the task of finding a reliable

structure easier. In so doing, we have added another dimension to the discussion around finding portable, or generalizable, results with ANNs relative to their ability to produce above market returns.

Our initial concerns with the portability of a model's returns were related to 1) the noted tendency of ANN predictors to weaken over time when applied to markets 2) the differences in returns produced by the B&H strategy over different date ranges that create more or less favorable return comparisons, and 3) demonstrating robustness to account drawdown periods (periods of poor return performance) should these be encountered early in a system's deployment. Here, we uncover a fourth consideration for models which make predictions more than a single period (day, week, month, year) forward. Such systems prove more reliable to the extent they generalize across each division of input/output pairings relative to the possible test sets within a given date range.

What if we had used monthly data? This would allow 4-week forward projections without having to skip periods and input/output pairings before making P&L calculations (and new trading decisions), say, on the first Friday of every month. It seems unlikely this would create a more reliable MLP based system. This is because we would likely be tailoring our predictive system to the return profile of input/output pairings that occur on the first Friday of every month, and we might get different returns if we tested it against start and end dates that occurred the second Wednesday of every month, for example. We would need to show such a system was profitable, on average, regardless of which

day of the month our period starts and ends. This is because, by training and testing using only subsets of all possible datasets while making returns the measure of performance, we are likely to end up with an ANN structure that is biased toward the return profile of that data subset, and this can happen even where the predictions themselves differ only slightly between subsets, because the outcomes of those predictions may differ significantly. This can be true even when the model is developed, as here, on a date range predating the range where tests were conducted. Model tweaks and input factor adjustments after poor performing tests can produce this bias where the dynamics producing the test datasets are not relatively stable. While the predictor may perform, according to traditional error measures, just as well over other subsets of input/output pairings, the return profile for these is very likely to be different than the one tested against. The construction of a model that performs well across all possible data subsets within a date range, and which does so without respect to where in the date range the tests begin, may indeed be an achievable task. It is, however, a higher bar than is typically set for this type of research.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Methodological Limitations and Future Work

Our emphasis on hit rate as a key performance measure and on the sign of MLP outputs for determining our trading decisions would seem to call for a model using binary classification rather than continuous output. While this approach would eliminate thresholding as a tool for determining actionable signals, it is hard not wonder if our results (particularly with DV) would not be better using binary classification. Unfortunately, Encog's classification functionality does not integrate well with the customized functionality built into our program. In particular, for purposes of preliminary testing we designed our program to be flexible in several ways relating to input selection and the number of periods forward that would be predicted. We additionally chose (out of necessity) to normalize data over sliding windows, and we incorporated flexibility in determining the range within which the data would be normalized. Encog's classification features are rather tightly coupled with both normalization procedures and data representation, and this provided significant obstacles to adding classification as an optional feature to our system.

We were initially concerned that our high-performing tests were the result of our structures having been biased toward one or more of the five subsets created by making

predictions five weeks forward. While this appears to be at least partially true, it would be more plausible as a complete explanation for the variability in our performance if our models were of a more sophisticated variety and had we used backpropagation with tailored initialization parameters to train them. In fact, predictions made between datasets offset by one week are often identical even when the returns resulting from those predictions are quite different due to the offset. Comparisons between high and low performing tests (on different subsets of the dataset) indicated that by far the source of variable performance in our tests stems from differences in the actual return outcomes related to different input/output pairings more than from variability in the predictions made by our MLPs over the various tests. Thus, to the extent bias was introduced, it favored the specific return profile of particular subsets of input/output pairings rather than binary hit/miss predictions that were overly tailored to that subset. As our models are so simple as to be almost generic, and because each suffers from the same subset related performance variability, we suspect that any bias of this type stems from our choices of training set length rather than model architecture.

Our fixed structure and static feature set is quite likely a too rigid model to perform well across all the possible partitions of such an extended dataset. Significant improvement to our methodology might be made by using adaptive methods for feature selection and architecture determination for each test window. [Swanson97] provides a good example of such a methodology for ANNs used to forecast several macroeconomic variables.

7.2 Conclusion

The Efficient Market Hypothesis states, in weak form, that above market returns are not systematically obtainable over time. Previous research with ANNs and market forecasting has shown them to be very good at modeling future price levels. However, the extent to which ANNs can model future returns with enough precision to undermine EMH has yet to be shown determinatively. While we can find examples in the literature which claim to do so, there is limited evidence that returns produced by such models may be generalized beyond the datasets upon which they were tested. Additionally, prior research indicates that statically trained models, while they may produce good returns for finite test periods, are subject to performance degradation over time.

Our Dynamic Training & Testing and Dynamic Validation models employ windowing, common with time series data, for both training and testing in order to adapt a predetermined architecture with randomized initial parameters to changes in the underlying processes by which market prices are generated. We attempt to minimize the problem of performance degradation related to statically trained MLPs with fixed architectures and to test for robustness to different market conditions as well as to changes in the favorability of benchmark comparisons. While Dynamic Validation proved partially successful in terms of our preferred performance measures (hit rates, returns produced above the B&H strategy) with respect to time, both models proved to be highly sensitive to variability in the training, validation and testing datasets. We demonstrated this by showing that moderate variability in the historical dataset used to

train an ANN structure, and/or very slight variability in the dataset used to test that structure's predictive capacity, may produce spectacularly different returns for the same trading system. Our ability to repeat these results over multiple runs on extended test sets with variably initialized model parameters showed that the variability in return performance stems more from features inherent to market returns than from incidental differences between ANN parameters.

We also demonstrated, with our Dynamic Training & Testing model, that even as ANN models may generate predictions which produce good returns over extended periods, these may not persist as the underlying dynamics generating market prices change. In so doing, we showed that even dynamic models may fail as their inputs, to the extent these remain constant, become less predictive of future returns or the relationships between inputs cease to be captured by a fixed model.

In short, this research demonstrated or confirmed several issues relative to the ability of an ANN model to produce returns that can generalize to future data:

1. Even as ANN models may generate predictions which produce good returns over extended periods, these may not persist as the underlying dynamics generating market prices change.
2. Accuracy (hit rate) and other standard error measures are not sufficient measures of a forecasting model's ability to produce above market returns.
3. Return performance may not generalize when a system is deployed during unfavorable market conditions.

4. Variability in the training and testing dataset may severely impact performance.
5. Variability in choice of input/output pairings for the model may produce dramatically differing return performance even as standard error measures remain relatively stable.

The results here leave EMH relatively unscathed. They are hardly definitive, however. While the methodologies employed here are somewhat distinctive for ANN research attempting to predict index returns, the ANN structures and trading strategy are rather rudimentary. More sophisticated models might do significantly better against variable datasets. However, such models may prove more credible to the extent they account for the kinds of variability described above.

REFERENCES

Print Publications:

[Asadi12]
Asadi, S., Hadavandi, E., Mehmanpazir, F., & Nakhostin, M. M. (2012). Hybridization of Evolutionary Levenberg–Marquardt Neural Networks and Data Pre-processing for Stock Market Prediction. Knowledge-Based Systems, 35, 245-258.

[Bachelier00]
Bachelier, L. (1900). Theory of Speculation, reprinted in P. Cootner (ed.), The Random Character of Stock Market Prices.

[Birgul03]
Birgul Egeli, A. (2003). Stock Market Prediction Using Artificial Neural Networks. Decision Support Systems, 22, 171-185.

[Brabazon02]
Brabazon, A. (2002). Financial Time Series Modeling Using Neural Networks: An Assessment of the Utility of a Stacking Methodology. In Artificial Intelligence and Cognitive Science (pp. 137-143). Springer Berlin Heidelberg.

[Brabazon06]
Brabazon, A., & O'Neill, M. (2006). Biologically Inspired Algorithms for Financial Modeling. Springer Science & Business Media.

[Brown95]
Brown, S. J., & Goetzmann, W. N. (1995). Performance Persistence. The Journal of Finance, 50(2), 679-698.

[Chen97]
Chen, S. H., & Yeh, C. H. (1997). Toward a Computable Approach to The Efficient Market Hypothesis: An Application of Genetic Programming. Journal of Economic Dynamics and Control, 21(6), 1043-1063.

[Clark12]
Clark, C., & Ranjan, R. (2012). How Do Proprietary Trading Firms Control the Risks of High Speed Trading? Chicago Fed Policy Document, 2012-1.

[Constantinou06]
Constantinou, E., Georgiades, R., Kazandjian, A., & Kouretas, G. P. (2006). Regime Switching and Artificial Neural Network Forecasting of The Cyprus Stock Exchange Daily Returns. International Journal of Finance & Economics, 11(4), 371-383.

[de Oliveira13]
de Oliveira, F. A., Nobre, C. N., & Zarate, L. E. (2013). Applying Artificial Neural Networks to Prediction of Stock Price and Improvement of The Directional Prediction Index–Case Study of PETR4, Petrobras, Brazil. Expert Systems with Applications, 40(18), 7596-7606.

[Diebold12]
Diebold, F. X., & Mariano, R. S. (2012). Comparing Predictive Accuracy. Journal of Business & Economic Statistics.

[Elman 90]
Elman, J. L. (1990). Finding Structure in Time. Cognitive Science, 14(2), 179-211.

[Faith07]
Faith, C. M., & Foster, M. (2007). Way of the Turtle. New York: McGraw-Hill.

[Fama95]
Fama, E. F. (1995). Random Walks in Stock Market Prices. Financial Analysts Journal, 51(1), 75-80.

[Fang14]
Fang, Y., Fataliyev, K., Wang, L., Fu, X., & Wang, Y. (2014, July). Improving the Genetic-Algorithm-Optimized Wavelet Neural Network for Stock Market Prediction. In 2014 International Joint Conference on Neural Networks (IJCNN) (pp. 3038-3042). IEEE.

[Fernandez-Rodriguez00]
Fernandez-Rodrıguez, F., Gonzalez-Martel, C., & Sosvilla-Rivero, S. (2000). On the Profitability of Technical Trading Rules Based on Artificial Neural Networks: Evidence from The Madrid Stock Market. Economics Letters, 69(1), 89-94.

[Franses98]
Franses, P. H., & Van Griensven, K. (1998). Forecasting Exchange Rates Using Neural Networks for Technical Trading Rules. Studies in Nonlinear Dynamics & Econometrics, 2(4).

[Gencay99]
Gencay, R. (1999). Linear, Non-Linear and Essential Foreign Exchange Rate Prediction with Simple Technical Trading Rules. Journal of International Economics, 47(1), 91-107.

[Goetzmann94]
Goetzmann, W. N., & Ibbotson, R. G. (1994). Do Winners Repeat? The Journal of Portfolio Management, 20(2), 9-18.

[Heaton08]
Heaton, J. (2008). Introduction to Neural Networks with Java. Heaton Research, Inc.

[Hornik89]
Hornik, K., Sinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. Neural networks, 2(5), 359-366

[Jarusqewicz04]
Jaruszewicz, M., & Mańdziuk, J. (2004). One day prediction of NIKKEI index Considering Information from Other Stock Markets. Artificial Intelligence and Soft Computing-ICAISC (pp. 1130-1135). Springer Berlin Heidelberg.

[Johnson10]
Johnson, B. (2010). Algorithmic Trading & DMA: An Introduction to Direct Access Trading Strategies. 4Myeloma Press.

[Jordan86]
Jordan, M. I. (1986). Attractor Dynamics and Parallelism in A Connectionist Sequential Machine. In Proceedings of the Eighth Annual Meeting of the Cognitive Science Society, pp. 531-546

[Kanas01]
Kanas, A., & Yannopoulos, A. (2001). Comparing Linear and Nonlinear Forecasts for Stock Returns. International Review of Economics & Finance, 10(4), 383-398.

[Kuan95]
Kuan, C. M., & Liu, T. (1995). Forecasting Exchange Rates Using Feed-forward and Recurrent Neural Networks. Journal of Applied Econometrics, 10(4), 347-364.

[Kuhn58]
Kuhn, H. W., & Tucker, A. W. (1958). John von Neumann's Work in The Theory of Games and Mathematical Economics. Bulletin of the American Mathematical Society, 64(Part 2), 100-122.

[Kuo98]
Kuo, R. J. (1998). A Decision Support System for the Stock Market through Integration of Fuzzy Neural Networks and Fuzzy Delphi. Applied Artificial Intelligence, 12(6), 501-520.

[LeBaron92]
LeBaron, B. (1992). Nonlinear forecasts for the S&P stock index. In Santa Fe Institute Studies in the Sciences of Complexity Proceedings (Vol. 12, pp. 381-381). Addison-Wesley Publishing Company.

[Leitch91]
Leitch, G., & Tanner, J. E. (1991). Economic Forecast Evaluation: Profits versus the Conventional Error Measures. The American Economic Review, 580-590.

[Lemperiere14]
Lempérière, Y., Deremble, C., Seager, P., Potters, M., & Bouchaud, J. P. (2014). Two Centuries of Trend Following.

[Levenberg44]
Levenberg, Kenneth. A Method for The Solution of Certain Non–Linear Problems in Least Squares. Quarterly of Applied Mathematics, 2(2), 164-168.

[Lewis14]
Lewis, M., & Baker, D. (2014). Flash Boys. Allen Lane.

[Lowenstein00]
Lowenstein, R. (2000). When Genius Failed: The Rise and Fall of Long-Term Capital Management. Random House Trade Paperbacks.

[Malkiel73]
Malkiel, B. G. (1973). A Random Walk Down Wall Street. WW Norton & Company.

[Malkiel99]
Malkiel, Burton Gordon (1999). A Random Walk Down Wall Street: Including A Life-Cycle Guide to Personal Investing. WW Norton & Company.

[Minsky69]
Minsky, M., & Papert, S. (1969). Perceptrons. MIT Press.

[Mizuno98]
Mizuno, H., Kosaka, M., Yajima, H., & Komoda, N. (1998). Application of Neural Network to Technical Analysis of Stock Market Prediction. Studies in Informatics and control, 7(3), 111-120.

[Murphy99]
Murphy, J. J. (1999). Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. Penguin.

[Neto10]
Neto, M. C. A., Tavares, G., Alves, V. M., Cavalcanti, G. D., & Ren, T. I. (2010, July). Improving Financial Time Series Prediction Using Exogenous Series and Neural Networks Committees. In The 2010 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[Neumann44]
Neumann, J. V., & Morgenstern, O. (1944). Theory of Games and Economic Behavior (Vol. 60). Princeton: Princeton University Press.

[Pan05]
Pan, H., Tilakaratne, C., & Yearwood, J. (2005). Predicting Australian Stock Market Index Using Neural Networks Exploiting Dynamical Swings and Intermarket Influences. Journal of Research and Practice in Information Technology, 37(1), 43-56.

[Perez-Rodriguez05]
Pérez-Rodríguez, J. V., Torra, S., & Andrada-Félix, J. (2005). STAR and ANN Models: Forecasting Performance on the Spanish "Ibex-35" Stock Index. Journal of Empirical Finance, 12(3), 490-509.

[Pesaran92]
Pesaran, M. H., & Timmermann, A. (1992). A Simple Nonparametric Test of Predictive Performance. Journal of Business & Economic Statistics, 10(4), 461-465.

[Regnault63]
Regnault, J. (1863). Calculation of Opportunities and Philosophy of The Bourse. Mallet-Bachelier.

[Riedmiller93]
Riedmiller, M., & Braun, H. (1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. Neural Networks, IEEE International Conference On. (pp. 586-591). IEEE.

[Rosenblatt58]
Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological review, 65(6), 386.

[Sinervo96]
Sinervo, B., & Lively, C. M. (1996). The Rock-Paper-Scissors Game and the Evolution of Alternative Male Strategies. Nature, 380(6571), 240-243.

[Soros03]
Soros, G. (2003). The Alchemy of Finance. Hoboken.

[Swanson97]
Swanson, N. R., & White, H. (1997). A Model Selection Approach to Real-Time Macroeconomic Forecasting Using Linear Models and Artificial Neural Networks. Review of Economics and Statistics, *79*(4), 540-550.

[Tsai09]
Tsai, C. F., & Chiou, Y. J. (2009). Earnings Management Prediction: A Pilot Study of Combining Neural Networks and Decision Trees. Expert Systems with Applications, 36(3), 7183-7191.

Electronic Sources:

[Goldstein15]
Goldstein, S. (2015). Here's the Map of the World, if Size Was Determined By Market Cap. http://www.marketwatch.com/story/heres-the-map-of-the-world-if-size-was-determined-by-market-cap-2015-08-12, last accessed: 12-2-2016.

[NYSE16]
(2016) NYSE Group Shares Outstanding and Market Capitalization of Companies Listed, 2016. http://www.nyxdata.com/nysedata/asp/factbook/viewer_edition.asp ?mode=tables&key=333&category=5, last accessed: 12-2-16.

[Quants13]
(2013) Quants: The Rocket Scientist of Wall Street. http://www.forbes.com/sites/investopedia/2013/06/07/quants-the-rocket-scientists-of-wall-street/#21573eb4444b, last accessed: 12-2-2016.

[Heaton08]
Heaton, J. http://www.heatonresearch.com/encog/, 12-2-16.

Appendix A

THE DATA

Then data used for model input are values derived from three datasets. These are weekly attribute values for the S&P 500, Brent Crude Oil futures and the US Dollar Index for Major Currencies. The S&P 500 dataset was downloaded from Yahoo Finance and includes weekly values from 1950 through 2016. The Brent and Dollar datasets were downloaded from the US Federal Reserve's FRED database. The Brent data contain dates and weekly closing prices from 1986 through 2016, and the US Dollar Index begins in 1973 and also contains weekly closing values through 2016. Not all data records are used, and those which are used depend on the dates chosen at runtime. Our program also provides the flexibility to choose attributes at runtime where the dataset contains more than one non-date attribute. Only those attributes which are required to compute the model inputs listed in Chapter 5 are used for the tests reported here, however.

All values input into the models, with the exception of the S&P 500 closing value on the date a prediction is made, are derived rather than raw, and these are then normalized prior to being input into the models. These derived values include the percentage changes and simple moving average ratios listed in Chapter 5. With the exception of the Bar Summary indicator, these are standardized computations, and they are made by the program at runtime. Each dataset was downloaded as a CSV file and then transformed

into an XML file to facilitate runtime processing. Processing each dataset at runtime, rather than consolidating the datasets into a flat file, provided flexibility during

preliminary testing with respect to attribute selection across multiple datasets and relative to the number of periods forward predicted. A side-by-side view of the first 15 values from each csv file can be seen in figure A1.

			S&P 500					Brent		US Dollar	
Date	Open	High	Low	Close	Volume	Adj Close		Date	Close	Date	Close
1/3/1950	16.66	16.98	16.66	16.98	1927500	16.98		1/3/1986	25.78	1/3/1973	108.2588
1/9/1950	17.08	17.09	16.67	16.67	2722000	16.67		1/10/1986	25.99	1/10/1973	108.3099
1/16/1950	16.72	16.9	16.72	16.9	1486000	16.9		1/17/1986	24.57	1/17/1973	108.3168
1/23/1950	16.92	16.92	16.73	16.82	1338000	16.82		1/24/1986	20.31	1/24/1973	108.2693
1/30/1950	17.02	17.29	17.02	17.29	1878000	17.29		1/31/1986	19.69	1/31/1973	107.8288
2/6/1950	17.32	17.32	17.21	17.24	1584000	17.24		2/7/1986	16.72	2/7/1973	107.5311
2/14/1950	17.06	17.15	16.99	17.15	1950000	17.15		2/14/1986	16.25	2/14/1973	107.087
2/20/1950	17.2	17.28	17.17	17.28	1425000	17.28		2/21/1986	14.39	2/21/1973	101.3312
2/27/1950	17.28	17.29	17.22	17.29	1398000	17.29		2/28/1986	14.25	2/28/1973	100.5567
3/6/1950	17.32	17.32	17.07	17.09	1402000	17.09		3/7/1986	12.27	3/7/1973	99.7613
3/13/1950	17.12	17.49	17.12	17.45	1538000	17.45		3/14/1986	13.07	3/14/1973	99.3603
3/20/1950	17.44	17.56	17.44	17.56	1686000	17.56		3/21/1986	13.45	3/21/1973	100.0502
3/27/1950	17.46	17.53	17.29	17.29	2010000	17.29		3/28/1986	12	3/28/1973	100.4555
4/3/1950	17.53	17.78	17.53	17.78	1752500	17.78		4/4/1986	11.44	4/4/1973	100.7945
4/10/1950	17.85	17.98	17.75	17.96	2250000	17.96		4/11/1986	13.46	4/11/1973	100.7113

Table 7: Raw CSV Data

Both the CSV and XML files used for our tests are available upon request.

VITA

Kevin Harper has a Bachelor's of Science from the University of Florida in Psychology and expects to obtain a Master in Computer Science from the University of North Florida, December, 2016. Dr. Sheriff Elfayoumy of the University of North Florida is serving as Kevin's thesis advisor. Kevin is currently employed at the University of North Florida as Support Technician at the College of Arts & Sciences. Prior to this position, Kevin spent 19 months with GE Transportation as a Software Developer Intern.

Kevin has experience programming with Perl, C, C# and Java programming languages, and he has written many complex SQL queries as a developer for GE. He is currently pursuing employment as .NET developer here in Jacksonville, FL.