Journal of Internet Banking and Commerce

An open access Internet journal (http://www.icommercecentral.com)

Journal of Internet Banking and Commerce, April 2018, vol. 23, no. 1

PREDICTING AND BEATING THE STOCK MARKET WITH MACHINE LEARNING AND TECHNICAL ANALYSIS

ANTHONY MACCHIARULO

Morgan Stanley and Co LLC NYC, NY, USA

Tel: **7185170966;**

Email: **macchiarulo.a@gmail.com**

Abstract

The paper studies whether machine learning or technical analysis best predicts the stock market and in turn generates the best return. The research back tests machine learning and technical analysis methods ten years in the past to predict ten years in the future. After prediction stage, the research incorporates the main findings into trading strategies to beat the S&P 500 index. To further this analysis, the paper examines all market periods and then examines the results specifically in up market and down-market periods. The sampling period is January 1995 through December 2005, and the trading period is January 2006 through December 2016. The null hypothesis is that machine learning and technical analysis would generate returns with no statistically significant difference. The study uses State Street's SPDR® SPY ETF as the benchmark. Data is retrieved from Bloomberg and Yahoo Finance. Outputs are calculated in R, MATLAB, SPSS, EVIEWS, Python, and SAS languages.

Keywords: **Machine Learning; Technical Analysis; Statistics; Predicting; Stock Market; Analysis; Investing; Trading; Securities**

INTRODUCTION

Machine Learning

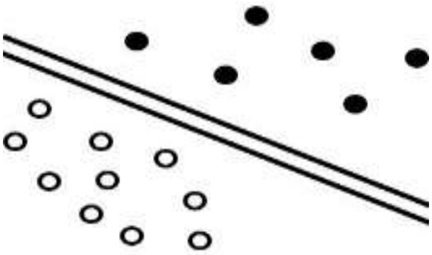
The inspiration for the machine learning portion of the research stems from the paper "Stock Price Prediction uses Neural Network with Hybridized Market Indicators" by Ayodele, et al. [1] Sunday published in the Journal of Computing. This paper focuses on predicting the stock market with machine learning techniques such as neural networks, support vector machines, and various other projects.

Machine Learning is a type of computational artificial intelligence that learns when exposed to new data. Machine Learning is used to predict the stock market. Some researchers claim that stock prices conform to the theory of random walk, which is that the future path of the price of a stock is not more predictable than random numbers. However, Stock prices do not follow random walks. There is sufficient evidence that shows that stock returns are predictable based on historical information. Three most prevalent Machine Learning Algorithms implemented in the field of finance are Support Vector Machines, Neural Networks, and Ensemble Learning. In the study, we use support vector machines to predict the relative direction of the stock market, and neural networks to predict the actual stock price and return. Ensemble learning allows us to combine the two machines into one prediction.

Support Vector Machine

Support Vector Machines increase the dimension of samples until it can linearly separate classes into a test set. Support Vector Machines use a mathematical formula known as the kernel function. The kernel function transforms the data so that there is a greater possibility of separable classes. When the machine has reached a state where it can linearly separate the classes, it attempts to find the optimal separation. When the machine has built its model, it can start to predict on new data by performing the same kernel transformation on the new data and decide what class it should belong to. The support vector machine creates a decision boundary where most points fall on either side of the boundary. The line in the support vector machine is known as the optimal hyper plane. A line is bad if it passes too close to the points because it will be too noise sensitive and it will not generalize correctly. Thus, the line passing as far as possible from all points is optimal. The standard formula for a hyper plane is $f(x)=\beta0 + \beta Tx$. $\beta0$ is referred to as the bias while βTx is the weight vector. The support vector uses Lagrange multipliers to obtain the weight and bias vector for the optimal hyper plane. Lagrange multiplier strategy attempts to find the local maximum and minimums of a function to equal constraints. The best implication for a support vector machine is to predict the direction of the stock market, that being either positive for negative in different market types such as a bear or bull market. The Figure 1, details linear separation with the kernel function.

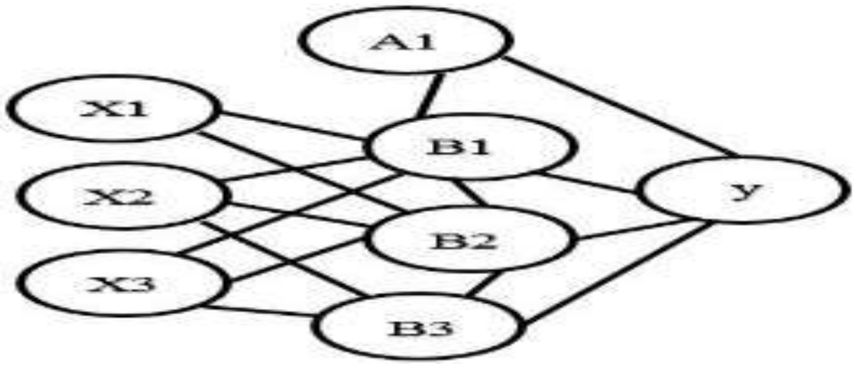
Figure 1: Support Vector Machine.



Neural Network

Neural networks take advantage of the way a biological brain solves problems with large clusters of biological neurons connected by axons in neither a way that a standard computer program cannot process nor a human process as efficiently. Neural Networks use a process called feed-forward backpropagation. The algorithm takes input variables and tries to predict the target variable. Neural Networks self-adjust input weights by testing millions of possibilities to optimize the target value to what is wanted by the user of the algorithm, whether it is a specified value, a prediction, or a maximization type of optimization problem. In our research, we will try to predict the stock market with the input variables. Trained data refers to the combination of input and target data. Neural network machines produce an R^2 of 0.99 if input and target data is consistent. An example of neural network is given below with three inputs, two hidden layers, and one target value (Figure 2).

Figure 2: Neural network.



Ensemble Learning

Ensemble Learning utilizes multiple learning algorithms to obtain better predictive powers. The learners are trained independently and predictions are combined to make the overall prediction. In our research, we will utilize ensemble learning to combine the results from the Neural Network and Support Vector Machines. Different techniques of ensemble learning relate to bootstrapping and stacking. Bagging or Bootstrap aggregating assigns equal weights to all the machines in the system. Stacking refers to separating algorithms and choosing the one with the best predictability. For our research stacking is the most efficient ensemble learning practice.

Noise

Noise is created from uncertainty and large impact events that can skew the machine learning process. The process of Cross validation is used to eliminate this from the model. Machine Learners attempt to build a model so that for a set of inputs, it can provide the wanted output. When the model emphasizes having low error too much, the model creates a decision boundary that is overly complicated and includes the noise. When the model allows for too great of an error, it is not able to properly divide the classes. To avoid the problems of over and under fitting; cross validation is used. Cross validation is a model evaluation method. Cross validation removes some of the data before training begins. When the training is done, the data that was removed is used to test the performance of the fitted model with unseen data.

Technical Analysis

The inspiration for the technical analysis portion of the research stems from the paper "Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support" by Leigh, et al. [2] published in the Journal of Finance. This paper focuses on predicting the stock market with technical analysis indicators as compared to neural network techniques of predicting the stock market.

As described in the paper, using technical analysis accepts a semi-strong form of the efficient markets hypothesis ("EMH"), which means that publicly available information about the stock should be factored into the stock price, and ignoring the weak form of EMH, which states that only past trading history has been built into the price. The paper examines the validity of the weak form of the EMH. In their comparison, they used a random-selection trading strategy to showcase the optimal weak EMH method. In their analysis, they took a series of price and volume patterns in different methods. They proved that the weak form EMH is not efficient in the face of momentum in stock prices. However, their most promising results were in the form of neural networks which are incorporated into the machine learning [3-6].

DATA AND METHODOLOGY

Machine Learning

The first step in the machine learning process to examine historical data that will be tested and define the sample and testing period. The sampling period is January 1995 through December 2005, and the trading period is January 2006 through December 2016. The next step in the Machine Learning process is to collect the data that will be used to predict the future of the stock market. In a machine, there is a set of data that contains both input data and target data, target data is the answer which the algorithm should produce from the input. These two sets of data combined are usually referred to as the training data. The training data is given below. By using previous data the machine should be able to predict the next years with precision (Table 1) [7-12].

Table 1: Input data.

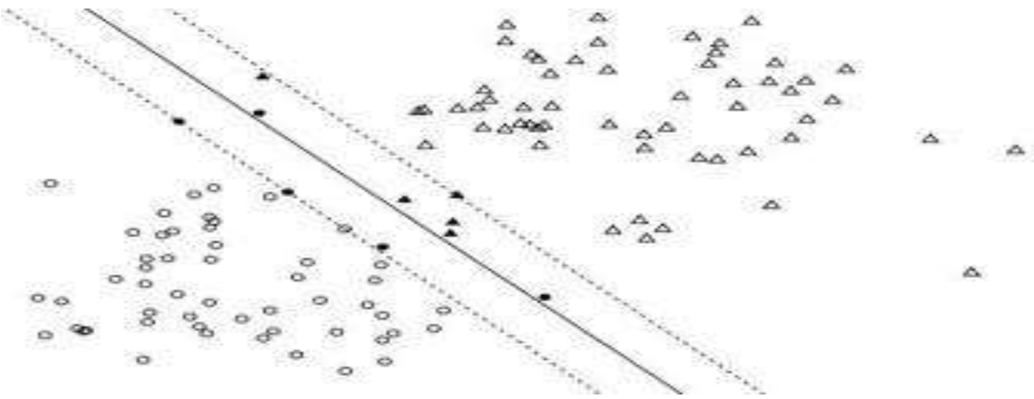
Driver	**Input Data**
S&P 500	Time
S&P 500	Open
S&P 500	High
S&P 500	Low
S&P 500	Close
S&P 500	Volume
Macroeconomic	United States 10 Yr. Treasury Bill
Macroeconomic	United States Inflation Rate
Macroeconomic	United States Unemployment Rate
Driver	Target Data
S&P 500	SPY Stock Price

Support Vector Machine

The next study that must be performed is the Support Vector Machine. We will be using the support vector machine to predict the market in both bull and bear trends. Using the input and target data we can fit the new model. The support vector machine asks for the number of data points and the number of dimensions. For the study, we will produce a set of positive and negative examples from two Gaussians. It is important to load standardized data such as sigma, the mean position, mean position for negative or bearish examples, and the mean position for bullish examples. Next the data must be trained. For the study, we split 80% into a training set and 20% into a test set. Using the kernel function, we predict the data points in the test set.

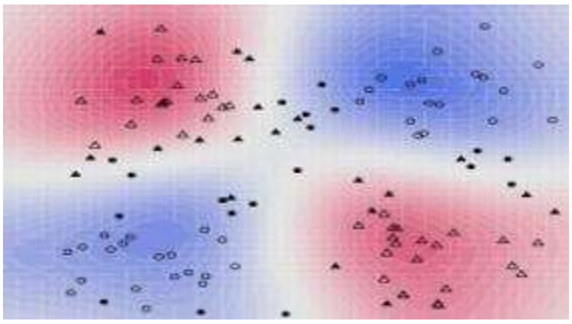
The dotted lines are the decision boundaries between positive and negative examples. The support vector is the black line. The triangle points above are the bullish scenario while the circle points below are the bearish scenario. The next step is to cross validate the training set to improve the quality of the machine and eliminate any noise. The k-fold and cross validation approaches are used by randomly splitting the number of samples into folds. Data is loaded into R. The Figure 3 is the linear support vector machine output.

Figure 3: Linear Support Vector Machine.



The linear support vector machine does not give all the information we need in predicting stock market direction. Just because we linearly separated positive or bullish and negative or bearish input parameters does not mean they are separable in real life. For example, if an economic rate falls that is considered a negative Gaussian but maybe the downward shift was a good sign for the economy. In the example of unemployment, if the unemployment rate decreases then that is good for the economy and is not accurately represented in the linear support vector machine. The nonlinear support vector machine tackles these problems in a more efficient manner. To transform the current machine into a nonlinear one we set the kernel parameter and a constant variable to one. Data is loaded into R, after running the nonlinear support vector machine, the results are shown in Figure 4.

Figure 4: Non-linear support vector machine.

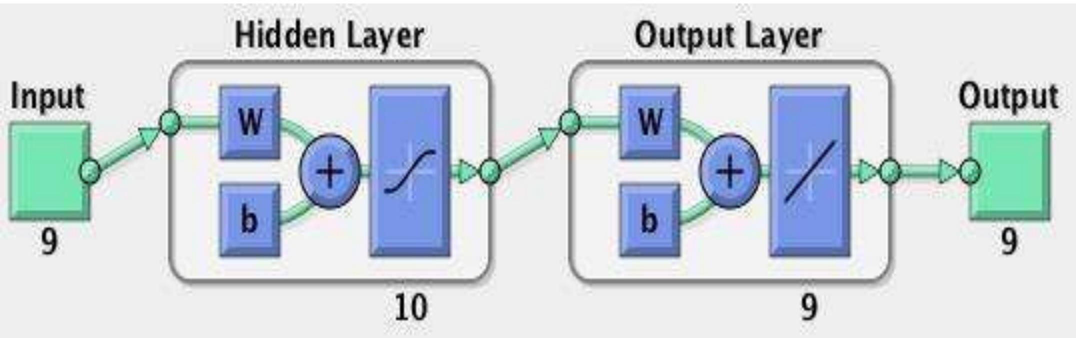


The linear and non-linear support vector machines tell the same conclusion in two different ways. For the linear support vector machine, there is more triangle or bullish points on the spectrum compared to bearish scenario. For the non-linear support vector machine, the bullish points are dispersed across the red heat map in much more quantities than the blue heat map. The darker red the heat map on the spectrum the more significance each point is making to the machine. In sum, this prediction dictates that there will be more bull trends than bear trends, which will make the stock market upward sloping and have a positive return for the trading period.

Neural Network

The next step is to fit the inputs and target into the neural network. The network developed will contain nine input variables with ten hidden layers. The target value or output in the neural network is the stock price in one year or the one-year return prediction for State Street's SPDR® SPY ETF ("SPY"). Data is loaded into MATLAB (Figure 5).

Figure 5: Neural network mapping.



Developing a neural network with external economic factors as inputs and the SPY stock price as output through feed-forward back propagation we assigned optimal weights to the individual SPY data and the external economic factors to not only predict the stock price in one year but also show the allocation of factors that lead to the prediction.

To remain consistent nine input and target values are distributed daily. 70% of the neural network is trained, 15% validated, and 15% tested. After training, cross validating, and testing the data the network runs and produces R^2 for each piece of the network. The R^2 for training, cross validation and testing is 0.99. The R^2 for the model is 0.97. This means that the neural network was performed correctly can be accepted with large confidence. The error histogram shows that the errors are normally distributed around the mean. Running the same simulation in R gives the same results. Using two independent packages increases the reliability of the study being conducted. Below are the results (Figure 6 and Table 2) [13-22].

Figure 6: Neural network training, validation, testing.

	Samples	MSE	R
Training:	4	2147483647.45...	9.99472e-1
Validation:	1	2147483647.66...	9.96173e-1
Testing:	1	2147483647.43...	9.99849e-1

Table 2: Neural network output.

Input Data	**Weight**
Time	3.43%
Open	7.52%
High	8.32%
Low	7.94%
Close	41.32%
Volume	25.11%
10 Yr. T-Bill	2.02%
Inflation	3.12%
Unemployment	1.22%
Target	Result
SPY Stock Price	117.16%

The neural network predicts the stock market at very high precision. The neural network in both studies yielded a ten-year return of 117.16% on the close of trading period. The neural network is only 1.04% below the actual return of 118.2%. That is very high predictability power. It is very interesting that the close price and volume of the SPY are the largest weights used by the network in determining the one year stock price. The external environmental factors play a much smaller role in the prediction determined by the network.

Machine Learning Trading Strategy

The next step is to develop the algorithm to trade based on the data. The support vector machine predicted the stock market to be upward sloping during the trading period and have a positive return. The support vector machine concludes this by dictating the number of bull and bear trends in the sample. With the support vector knowledge in mind running the neural network on the data predicted the stock market at a 1.04% margin of error. This is extremely high precision. In sum, the machine learning process has predicted that there will be more bull days than bear days and almost perfectly predicted the stock market. This type of knowledge is very powerful and useful to profit in finance.

When doing prediction, the close price and volume of the SPY are the largest weights used by the network in determining the one-year stock price. The external environmental factors play a much smaller role in the prediction determined by the network. Due to this discovery, the algorithm trades heavily based on lagged close prices and trading volume to maximize returns on the stock market. The algorithm trades by only rebalancing stocks in the S&P500 that are “winners” the day before that is a stock that ended positively the day before to incorporate the Support Vector Machine into the trades. Additionally, the rotation system does not execute rebalancing trades without there being larger volume compared to the stock’s average daily trading volume the day before. The results beat the S&P500 index as seen below. Additionally, we run a neural network in R for every previous period and if there was a larger weight given to closing price over trading volume we tweak the algorithm to check for close prices over trading volume 60% of the time as opposed to a 50/50 split. The vise-versa is true when trading volume was higher where we would trade on volume 60% of the time over close prices. The trading results are shown below. The algorithm is shown below before tweaking weights due to neural network parameter [23-30].

```
def initialize(context):
#constants
context.volu
me=0.5
context.clos
e=0.5
context.closed=data.history(sid(8554),
'price',                    1,                    '1d')
```

```
context.vol=data.history(sid(8554),
'volume', 1, '1d')
#ETF traded with weight
if context.vol > context.vol -1 and context.closed >
context.closed -1 then context.etfs={
  symbol('SPY'): 1.0, # State Street's SPDR® SPY ETF
}
end if
# Set commision
set_commission(commission.PerShare(cost=4.95, min_trade_cost=0.0))
# Rebalance portfolio
schedule_function(rebalance,date_rules.every_day(),
time_rules.market_open(minutes=35))
def rebalance (context, data):
for stock, weight in context.etfs.items():
order_target_percent(stock, weight*context.volume + weight*context.close)
```

The total return for the period is 204% as opposed to the S&P500 returns of 118.2%. The strategy beats the market on the long term as well. 69 times the machine learning strategy beats the market on a month to month basis out of 132 months. 52.27% of the time the strategy beats the markets monthly returns. The max drawdown of the strategy comes out to 46.9% during the recession. It is apparent the strategy does much better in a bullish market compared to a bearish market.

Running the strategy over ten years only produces a Beta of 0.72, which is less risky than investing in the market. Additionally, the Sharpe ratio is 0.51 and a Sortino negatively skewed at 0.71, and a volatility or standard deviation of 0.28. During the recession, the month with the highest beta was 2.598 during April 2007. This is expected and is much less risky than the market was during the time. In sum, the machine learning algorithm that learns based on the previous year and adjusts the strategy on percentage of buy and short based on trading volume and close prices beats the market by 85.8% over ten years with slightly higher volatility than the market. The strategy is more volatile 116 months out of the 131 months or 88.54% of the time the standard deviation of the strategy is higher than the market. For the higher volatility, the strategy to beat the market by almost doubles [31-34].

Technical Analysis

For each method, there were 120 total observations over the total sample period from January 2007 to December 2016. Machine learning had the highest overall average monthly return at 1.19%. During this same time-period, the S&P 500 had an average monthly return of .48%. The monthly average returns for the technical indicators ranged from .83% to -1.21%. The full listing of the average monthly returns listed in percent form is shown in Table 3.

Table 3: Whole sample period descriptive statistics (data in percent form).

	N	Minimum	Maximum	Mean	Std. Deviation
Machine Learning	120	-20.4	23.5	1.192917	7.347258
Bollinger Bands	120	-13.129	19.71627	0.831313	3.758738
Trading Envelopes	120	-13.129	19.71627	0.831313	3.758738
KBand	120	-13.129	13.31265	0.76489	4.25558
Cmdty Channel Index	120	-16.5331	13.06419	0.538209	3.622277
Stochastics	120	-9.02627	11.50051	0.492497	3.374356
William's %R	120	-9.02627	12.64072	0.408589	3.396424
Buy and Hold	120	-16.5331	14.2041	0.403511	4.553905
Fundamental Analysis	120	-18.46	10.18	0.383	4.45161
MA Envelopes	120	-6.10397	13.94343	0.289602	2.477517
RSI	120	-4.61304	12.65053	0.276992	1.865896
MACD	120	-9.07901	8.290536	0.255896	3.016252
Ichimoku	120	-5.92016	7.675862	0.050554	2.08152
Triangular MA	120	-8.58764	6.438574	-0.1277	2.515717
DMI	120	-14.0814	8.636103	-0.19479	2.808664
Exponential MA	120	-8.7846	8.829373	-0.22599	2.59998
MA Oscillator	120	-10.4149	10.87842	-0.23029	3.917283
Fear and Greed	120	-13.9833	9.543643	-0.23482	3.406039
Simple MA	120	-16.8559	8.496324	-0.54573	3.442464
Weighted MA	120	-15.3914	6.827461	-0.55405	3.148252
Variable MA	120	-23.4974	6.569704	-0.67838	3.79582
Parabolic	120	-23.3883	11.8544	-0.69678	4.676691
Accum/Distrib Osc	120	-21.6401	16.67208	-1.01638	5.40478

Rex Oscillator	120	-18.6651	11.25333	-1.02282	4.683956
Rate of Change	120	-18.1407	14.31845	-1.20797	4.95222
Valid N (list wise)	120				

After gathering the sample period data, we separated out the observations into those that occurred in an up market from those in a down market. This was done by looking at the returns of the S&P 500. For months when it was positive, the returns for that month were classified as up market and when it was negative; the returns were classified as down market. The up-market period had a total of 72 observed months. During this time, the S&P 500 had an average monthly return of 3.22%. Machine learning had 4.13% monthly average return, approximately 1% above the next highest method. As seen in Table 3, the technical indicators ranged from 2.99% to -1.01% (Table 4).

Table 4: Up market descriptive statistics (data in percent form).

		Minimum	**Maximum**	**Mean**	**Std. Deviation**
Machine Learning	72	-8.2	23.5	4.1289	5.96786
Fundamental Analysis	72	0.02	10.18	3.1292	2.30648
Buy and Hold	72	-2.49769	14.2041	2.988633	2.976227
Bollinger Bands	72	-2.44045	19.71627	1.092965	3.311484
Trading Envelopes	72	-2.44045	19.71627	1.092965	3.311484
Cmdty Channel Index	72	-4.63839	7.445843	0.391186	2.841843
RSI	72	-2.83878	12.65053	0.31323	2.091257
Stochastics	72	-5.28675	11.50051	0.296028	3.367026
Fear and Greed	72	-7.86461	9.543643	0.164008	2.801145
Triangular MA	72	-6.90072	4.913777	0.110924	2.06229
Ichimoku	72	-5.52048	4.376417	0.069515	1.529319
KBand	72	-6.10397	9.010012	0.060722	3.453896

Exponential MA	72	-8.7846	8.829373	0.056914	2.64089
MA Envelopes	72	-6.10397	8.57509	0.052518	1.949429
MACD	72	-6.44503	8.215632	0.041862	2.892891
William's %R	72	-6.86243	9.890922	0.034897	2.869658
DMI	72	-6.88233	4.667657	0.00081	2.372219
Parabolic	72	-9.11704	10.87842	-0.12789	3.630527
Rex Oscillator	72	-15.5689	11.25333	-0.30996	4.202107
Simple MA	72	-16.8559	8.496324	-0.3947	3.452201
Weighted MA	72	-15.3914	4.913777	-0.45	3.179792
MA Oscillator	72	-10.4149	10.87842	-0.61118	3.596882
Variable MA	72	-23.4974	6.569704	-0.63494	4.044417
Accum/Distrib Osc	72	-17.8133	8.257169	-0.89352	3.753331
Rate of Change	72	-17.8133	5.857087	-1.00538	4.130339
Valid N (list wise)	72				

For the down market, as seen below in Table 5, we only had a total of 48 observations. During the time, the S&P 500 had an average monthly return of -3.63%. Machine learning did not perform as well as in the whole sample and up market periods and had -3.21% for its monthly average return. However, the technical indicators were more varied ranging between 1.82% to -3.47% (Table 5).

Table 5: Down market descriptive statistics (data in percent form).

	N	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**
KBand	48	-13.129	13.31265	1.821142	5.092432
William's %R	48	-9.02627	12.64072	0.969127	4.028968
Stochastics	48	-9.02627	8.703512	0.797199	3.399316

Cmdty Channel Index	48	-16.5331	13.06419	0.758744	4.575824
MA Envelopes	48	-5.4208	13.94343	0.645228	3.095968
MACD	48	-9.07901	8.290536	0.576946	3.196419
Bollinger Bands Trading Envelopes MA Oscillator	48	-13.129	15.85366	0.438836	4.352406
	48	-13.129	15.85366	0.438836	4.352406
	48	-9.04289	10.24691	0.341037	4.330465
RSI	48	-4.61304	4.268927	0.222634	1.484401
Ichimoku	48	-5.92016	7.675862	0.022111	2.726831
Triangular MA	48	-8.58764	6.438574	-0.48563	3.062873
DMI	48	-14.0814	8.636103	-0.49819	3.365372
Expoential MA	48	-8.53862	6.070957	-0.65036	2.504656
Weighted MA	48	-9.08872	6.827461	-0.71012	3.127204
Variable MA	48	-8.83962	5.930361	-0.74354	3.429761
Simple MA	48	-9.56601	5.930361	-0.77227	3.451656
Fear and Greed Accum/Distrib Osc	48	-13.9833	7.076658	-0.83306	4.112265
	48	-21.6401	16.67208	-1.20066	7.254137
Rate of Change	48	-18.1407	14.31845	-1.51186	6.01379
Parabolic	48	-23.3883	11.8544	-1.55011	5.850289
Rex Oscillator	48	-18.6651	10.76828	-2.09211	5.189241
Machine Learning	48	-20.4	14.19	-3.211	7.06144
Buy and Hold	48	-16.5331	2.241661	-3.47417	3.678578

Fundamental Analysis	48	-18.46	-0.1	-3.7363	3.64085
Valid N (listwise)	48				

RESULTS

To test for statistical significance for the machine learning results compared to those of the technical analysis, we used paired samples t-tests. The results, as seen below in Table 6, are ordered from the highest average monthly return to the lowest for each of the technical indicators, compared to the machine learning results which had the highest mean. At a 95% confidence level, machine learning outperformed the following technical indicators: fear and greed, simple MA, weighted MA, variable MA, parabolic, accum/distrib osc, Rex Oscillator, and rate of change. For the up-market period, machine learning had outperformed technical analysis results by a relatively large margin. As seen in Table 7 below, the results for the up-market period were better than those from the total 120 observations. At the 99% confidence level, machine learning outperformed compared to all but the buy and hold technical analysis method. Those two it outperformed with marginal significance at the 80% level. Compared to the results from the whole sample, this indicates that machine learning will be more likely to outperform in an up-market period.

Table 6: Paired t-test results (entire period, data in percent form).

Pair	Strategy	Mean	Std. Dev.	Std. Error Mean	Lower (95%)	Upper (95%)	T	Df	Sig (two tailed)
Pair 1	Bollinger Bands – Machine Learning	-0.36	8.2	0.74	-1.84	1.21	-0.48	119	0.63
Pair 2	Trading Envelopes – Machine Learning	-0.36	8.2	0.74	-1.84	1.12	-0.553	119	0.63
Pair 3	KBand – Machine Learning	-0.42	8.4	0.77	-1.96	1.104	-0.873	119	0.581
Pair 4	Cmdty Channel Index – Machine Learning	-0.654	8.2	0.74	-2.139	0.733	-0.967	119	0.384
Pair 5	Stochastics – Machine Learning	-0.7	7.9	0.72	-2.31	0.751	-1.01	119	0.335
Pair 6	Williams %R – Machine Learning	-0.78	8.49	0.77	-1.95	0.37	-1.34	119	0.314
Pair 7	Buy and Hold – Machine Learning	-0.8099	6.4	0.58	-1.99	0.29	-1.46	119	0.181

Pair 8	Fundamental Analysis – Machine Learning	-0.9033	6.09	0.55	-2.139	0.53	-1.24	119	0.148
Pair 9	MA Envelopes – Machine Learning	-0.915	7.94	0.77	-2.139	0.53	-1.3	119	0.215
Pair 10	RSI – Machine Learning	-0.93	7.68	0.74	-2.139	0.47	-1.19	119	0.194
Pair 11	MACD – Machine Learning	-1.14	8.68	0.74	-2.139	0.62	-1.776	119	0.236
Pair 12	Ichimoku – Machine Learning	-1.32	7.54	0.68	-3.009	0.22	-2.077	119	0.1
Pair 13	Triangular MA – Machine Learning	-1.38	8.13	0.72	-2.78	0.15	-2.32	119	0.078
Pair 14	DMI – Machine Learning	-1.41	8.12	0.74	-3.21	0.081	-2.3	119	0.064
Pair 15	Exponential MA – Machine Learning	-1.42	7.9	0.69	-3.41	0.01	-2.41	119	0.052
Pair 16	MA Oscillator – Machine Learning	-1.42	8.77	0.741	-3.8	0.16	-1.46	119	0.078
Pair 17	Fear and Greed – Machine Learning	-1.73	8.1	0.74	-3.667	-0.066	-1.24	119	0.04
Pair 18	Simple MA – Machine Learning	-1.74	8.3	0.74	-3.891	-0.25	-1.3	119	0.022
Pair 19	Weighted MA – Machine Learning	-1.14	8.4	0.722	-3.009	-0.24	-2.32	119	0.023
Pair 20	Variable MA – Machine Learning	-1.87	8.03	0.743	-2.78	-0.33	-2.3	119	0.017
Pair 21	Parabolic– Machine Learning	-1.88	8.241	0.74	-3.21	-0.36	-2.41	119	0.016
Pair 22	Accum/Distrib Osc. – Machine Learning	-2.02	8.805	0.74	-3.41	-0.61	-2.79	119	0.007
Pair 23	Rex Oscillator – Machine Learning	-2.21	8.031	0.74	-1.99	-0.76	-3.02	119	0.003
Pair 24	Rate of Change – Machine Learning	-2.4	8.24	0.734	-3.891	-0.9098	-3.11	119	0.002

The results for the down-market period showcased the weakness of machine learning. Although it performed above many technical indicators in the positive return period, it underperformed in the down-market period. Over the 48 observed months with a negative S&P 500 return, machine learning was close to being the lowest average monthly returns (Table 7).

Table 7: Up market paired samples t-test (data in percent form).

Pair	Strategy	Mean	Std. Dev.	Std. Error Mean	Lower (95%)	Upper (95%)	T	Df	Sig (two tailed)
Pair 1	Bollinger Bands – Machine Learning	-0.9997	5.675	0.66	-2.33	0.333	-1.4	71	0.139
Pair 2	Trading Envelopes – Machine Learning	-3.035	6.78	0.79	-4.62	-1.44	-3.7	71	0.118
Pair 3	KBand – Machine Learning	-3.034	6.857	0.79	-4.6	-1.44	-3.7	71	0
Pair 4	Cmdty Channel Index – Machine Learning	-3.73	6.42	0.802	-5.33	-2.13	-3.7	71	0
Pair 5	Stochastics – Machine Learning	-3.81	5.93	0.76	-6.72	-2.05	-3.7	71	0
Pair 6	Williams %R – Machine Learning	-3.82	6.87	0.808	-6.12	-2.04	-4.6	71	0
Pair 7	Buy and Hold – Machine Learning	-3.96	6.69	0.74	-4.62	-2.66	-4.7	71	0
Pair 8	Fundamental Analysis – Machine Learning	-4.01	5.93	0.75	-4.6	-2.45	-5.3	71	0
Pair 9	MA Envelopes – Machine Learning	-4.07	5.93	0.69	-5.33	-2.05	-5.8	71	0
Pair 10	RSI – Machine Learning	-4.09	7.17	0.809	-4.6	-2.04	-5	71	0
Pair 11	MACD – Machine Learning	-4.09	6.77	0.76	-6.72	-2.66	-5.1	71	0
Pair 12	Ichimoku – Machine Learning	-3.96	6.96	0.78	-6.12	-2.66	-5.5	71	0
Pair 13	Triangular MA – Machine Learning	-4.01	6.42	0.74	-6.11	-2.66	-5.6	71	0
Pair 14	DMI – Machine Learning	-4.07	5.93	0.801	-6.43	-2.45	-5.7	71	0
Pair 15	Exponential MA – Machine Learning	-4.07	6.87	0.74	-6.47	-2.45	-5.5	71	0
Pair 16	MA Oscillator – Machine Learning	-4.09	6.42	0.78	-6.72	-2.04	-5.6	71	0
Pair 17	Fear and Greed – Machine Learning	-4.09	6.42	0.74	-6.9	-2.66	-5.7	71	0
Pair 18	Simple MA – Machine Learning	-4.07	5.93	0.801	-6.11	-2.45	-5.5	71	0

Pair 19	Weighted MA – Machine Learning	-4.07	6.87	0.74	-6.43	-2.04	-5.6	71	0
Pair 20	Variable MA – Machine Learning	-4.09	6.78	0.78	-6.47	-2.66	-5.7	71	0
Pair 21	Parabolic– Machine Learning	-4.09	6.857	0.74	-6.72	-2.04	-5.7	71	0
Pair 22	Accum/Distrib Osc. – Machine Learning	-3.81	6.42	0.801	-6.47	-2.66	-5.5	71	0
Pair 23	Rex Oscillator – Machine Learning	-5.022	7.26	0.855	-6.729	-3.31	-5.8	71	0
Pair 24	Rate of Change – Machine Learning	-5.13	7.51	0.8855	-6.9	-3.36	-5.7	71	0

At a 95% confidence level, machine learning underperformed compared to the following technical analysis methods: KBand, William's %R, Stochastics, Cmdty Channel Index, MA Envelopes, MACD, Bollinger Bands, Trading Envelopes, RSI, Ichimoku, Triangular MA, DMI, Exponential MA, Weighted MA, Variable MA and Fear and Greed. With a marginal significance of 20%, machine learning significantly unperformed compared to Simple MA, Accum/Distrib OSC, and Rate of Change (Table 8).

Table 8: Down market paired samples t-test.

Pair	**Strategy**	**Mean**	**Std. Dev.**	**Std. Error Mean**	**Lower (95%)**	**Upper (95%)**	**T**	**Df**	**Sig (two tailed)**
Pair 1	Bollinger Bands – Machine Learning	5.03	7.74	1.118	3.15	6.9	4.5	47	0
Pair 2	Trading Envelopes – Machine Learning	4.1	8.45	1.22	2.13	6.22	3.4	47	0.001
Pair 3	KBand – Machine Learning	3.99	7.11	1.02	2.27	5.7	3.8	47	0
Pair 4	Cmdty Channel Index – Machine Learning	3.96	8.921	1.28	1.62	5.94	3.48	47	0.001
Pair 5	Stochastics – Machine Learning	3.64	7.74	1.118	3.15	6.9	3.22	47	0.001
Pair 6	Williams %R – Machine Learning	3.78	7.11	1.02	2.27	5.7	3.8	47	0.005
Pair 7	Buy and Hold – Machine Learning	2.37	8.921	1.28	1.62	5.94	3.48	47	0.005
Pair 8	Fundamental Analysis –	2.01	8.57	1.237	1.57	5.72	2.95	47	0.005

	Machine Learning								
Pair 9	MA Envelopes – Machine Learning	2.69	7.74	1.118	3.15	6.9	2.95	47	0.005
Pair 10	RSI – Machine Learning	3.99	7.11	1.02	2.27	5.7	3.8	47	0.005
Pair 11	MACD – Machine Learning	3.96	8.921	1.28	1.62	5.94	3.48	47	0.002
Pair 12	Ichimoku – Machine Learning	3.64	7.74	1.118	3.15	6.9	3.45	47	0.037
Pair 13	Triangular MA – Machine Learning	2.21	8.08	1.02	1.57	4.787		47	0.347
Pair 14	DMI – Machine Learning	2.23	8.24	1.28	1.66	4.85	3.22	47	0.231
Pair 15	Exponential MA – Machine Learning	2.27	8.4	1.237	0.58	4.873	3.8	47	0.783
Pair 16	MA Oscillator – Machine Learning	3.99	8.37	1.118	0.46	4.53	3.48	47	0.581
Pair 17	Fear and Greed – Machine Learning	3.96	8.412	1.213	0.401	4.49	2.95	47	0.005
Pair 18	Simple MA – Machine Learning	3.64	7.68	1.02	0.517	4.47	1.499	47	0.005
Pair 19	Weighted MA – Machine Learning	3.99	9.29	1.28	-0.23	4.23	1.541	47	0.006
Pair 20	Variable MA – Machine Learning	3.96	7.63	1.237	-0.85	4.26	0.95	47	0.47
Pair 21	Parabolic– Machine Learning	3.64	9.47	1.118	-1.19	3.95	-0.264	47	0.783
Pair 22	Accum/Distrib Osc. – Machine Learning	1.1	8.15	1.12	-0.85	3.09	-2.64	47	0.581
Pair 23	Rex Oscillator – Machine Learning	-0.26	6.89	0.99	-1.932	1.406	-0.261	47	0.006
Pair 24	Rate of Change – Machine Learning	-0.5252	6.7	0.97	-2.15	1.1	-0.541	47	0.47

CONCLUSION

In conclusion, after analyzing the results, we conclude that using machine learning as a trading strategy can positively impact the returns generated compared to using many technical indicators. We found that there was no statistically significant difference between using machine learning and using technical analysis. In up market periods, machine learning will outperform technical analysis. However, if the market is a down market it is more beneficial to use technical analysis. Machine

Learning performs better in up markets because it uses momentum to its advantage by calculating the optimal weights that need to be traded on in the market paired with the future direction. On the other hand, technical analysis performs much better at spotting potential drawdowns, especially when using so many different trading strategies it is apparent some work better than others in down markets. For future research, we would recommend examining similar methods over a longer time-period. Because the down market only had 48 observations, it might have decreased the usability of the results.

REFERENCES

1. Ayodele A, Charles A, Marion A, Sunday O. Stock price prediction using neural network with hybridized market indicators. The Journal of Computing, pp: 2-54.
2. Leigh W, Purvis R, Ragusa JM. Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in romantic decision support. Journal of Finance, pp: 10-112.
3. Abarbanell J, Bushee B. Fundamental analysis, future earnings, and stock prices. Journal of Accounting Research, pp: 8-45.
4. Freeman JA, Skapura DM (1991) Neural networks: algorithms, applications and programming techniques. Addison Wesley Longman, pp: 18-44.
5. Vikas A, Naik NY (2004) Risk and portfolio decisions involving hedge funds. Review of Financial Studies 17: 63-98.
6. Karazoglou A. Supper vector machines in R. Journal of statistical software, pp: 12-28.
7. Andrew A, Gorovyy S, van Inwegen GB (2011) Hedge fund leverage. Journal of Financial Economics 102: 102-126.
8. Heckerling PS, Canaris G, Flach SD, Tape TG, Wigton RS,et al. (2007) Predictors of urinary tract infection based on artificial neural networks and genetic algorithms. International Journal of Medical Informatics 76: 289-296.
9. Watkins C (2000) Dynamic alignment kernels. In: Smola A, Bartlett PL, Sch¨olkopf B, Schuurmans D (eds.), Advances in large margin classifiers. MIT Press, Cambridge, MA, pp: 39-50.
10. Clifford A, Krail R, Liew J (2001) Do hedge funds hedge? Journal of Portfolio Management 28: 6-19.
11. Ali A, Magnor O, Schultalbers M (2009) Misfire detection using a neural network based pattern recognition. International Conference on Artificial Intelligence and Computational Intelligence.
12. Anand MV (1994) S&P 500 trading strategies and stock betas. Review of Financial Studies 7: 215-251.

13. Dimitri V, Woolley P (2013) An institutional theory of momentum and reversal. Review of Financial Studies 26: 1087-1145.
14. Dybowski R, Gant V (2007) Clinical applications of artificial neural networks. Cambridge University Press, pp: 2-33.
15. Gil D, Johnsson M, Garicia Chemizo JM, Paya AS, Fernandez DR (2009) Application of artificial neural networks in the diagnosis of urological dysfunctions. Expert Systems with Applications 36: 5754-5760.
16. Wang Y-G, Li H-P (2010) Remote sensing image classification based on artificial neural network. International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE).
17. R Development Core Team (2008) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.
18. ISO/IEC (2014) ISO International Standard ISO/IEC 14882:2014(E) - Programming Language C++. Working draft, Geneva, International Organization for Standardization (ISO), Switzerland.
19. Van Rossum G. Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995. Python Software Foundation. Python Language Reference, version 2.7.
20. MATLAB and Signal Processing Toolbox Release 2012. The Math Works, Inc., Natick, Massachusetts, United States.
21. IBM Corp (2013) IBM SPSS Statistics for Windows, Version 22.0. Armonk. IBM Corp, NY.
22. SAS Institute Inc. Cary, NC, 1989-2007.
23. Friedrich A, Lutz VC (1950) Monetary and foreign exchange policy in Italy.
24. Schlesinger ER (1952) Multiple exchange rates and economic development.
25. White WR (2000) What have we learnt from recent financial crises and policy responses? BIS Working Papers.
26. Bloomberg (2012) Bloomberg Professional.
27. CRSP/Compustat Merged (2012) CRSP/Compustat Merged. Center for Research in Security Prices.
28. Thomson One Banker (2012) Thomson One Banker. Thomson Reuters.
29. Bloomberg LP (2006) Stock price graph for SPDR® SPY ETF.
30. Widrow B, Lehr M (1990) 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. Proceedings of the IEEE 78: 1420-1442.
31. Anderson J, Rosenfeld E (1988) Neurocomputing: Foundations of Research. MIT Press, Cambridge, MA.
32. Arrowsmith D, Place C (1990) An introduction to dynamical systems. Cambridge University Press, Cambridge, UK.
33. Dittman DJ, Khoshgoftaar TM, Napolitano A. Selecting the appropriate ensemble learning approach for balanced bioinformatics data.

34. Khoshgoftaar TM, Dittman DJ, Wald R, Fazelpour A (2012) First order statistics based feature selection: A diverse and powerful family of feature selection techniques. Proceedings of the Eleventh International Conference on Machine Learning and Applications (ICMLA): Health Informatics Workshop, pp: 151-157.