Forecasting S&P 500 Trend Using Inversely Correlated Assets

Hyeree Oh, Cameron Way, Kourosh Azizi
Simon Fraser University, Burnaby, Canada

Abstract

Financial markets are the center point of today's economy, and correctly predicting the future of an asset is a very lucrative venture. With stock prices being so volatile, they can be hard to predict. In this paper we use Support Vector Machines, Artificial Neural Networks, and Random Forests to show the correlation between volatility measures and inverse assets to predict next day price direction and trends of the S&P 500. We were able to predict next day trends of SPX with accuracy of 84% even when our model is given no data about SPX.

Keywords: support vectors, neural networks, random forests, machine learning, deep learning

Author roles: Hyeree: Code and Optimization, Cameron: Code and Data Analysis, Kourosh: Quantitative Strategy Development and Backtesting

1 Introduction

The global economy is essential to the modern world, and is greatly impacted by the financial markets. The performance of these markets can impact many things such as jobs, retirement savings, currencies, and worldwide trade. Throughout history, there have been major events that have caused extreme volatility in the financial markets. Volatility usually occurs when there is uncertainty about the future, the economy, tension between countries, or major events such as pandemics.

When the economy show signs of uncertainty, institutions tend to seek safety by liquidating their assets to hold more cash. In the market crash of 2022, institutions sold equities and bonds due to the raise in U.S. interest rates as a result of the growing inflation. The increase in interest rates means the cost of borrowing has gone up, so companies start cancelling projects and laying off employees. As a result, capital will flow out of equities into other areas. Fixed-income is a possible destination for this capital unless interest rates are increasing because the bonds purchased will be worth less once the rates go up. As selling starts in the bond market similar to equities, the U.S. Treasury Bond Yields increase in value in the hope of attracting buyers. Similarly, as investors hold more cash, the U.S. dollar also increases in value.

The transition of capital across these markets is easier said than done; the challenging part is mainly liquidity, especially when the capital is large. To hedge or move huge amounts of capital, institutions need time to react. That is why they have been focused on developing quantitative solutions that can predict trends ahead of a major event. Luckily, there are patterns and indicators that can help with identifying these trends at an early stage. A trend is the average of the closing price of various days grouped together. There can be an upward or a downward trend with a mix of bullish and bearish days. This paper ignores the direction of individual days, but rather focuses on the direction of various days grouped together. This approach can help us ignore anomalies and random price movements.

To predict the trend of S&P 500 index in a unique way, we will be using assets that are inversely correlated to index's performance. These assets include Volatility Index (VIX), U.S. Dollar Index (DXY), and U.S. Treasury Bond Yields (US10Y). We will be using machine learning and deep learning models to find out which model predicts the trend more accurately.

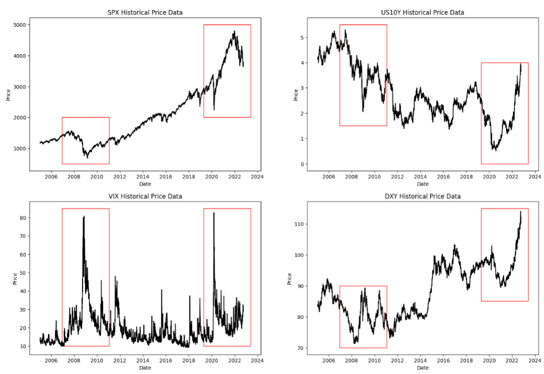


Figure 1: Historical Price of SPX(up left), US10Y(up right), VIX(down left), and DXY(down right)

There are certain economic conditions that build patterns among these assets as seen in Figure 1. This paper will try various algorithms to see which one performs the best at predicting trends more accurately. We hope this paper can ignite new ideas that can be stepping stones for us or others to explore in the future.

SVM is a method of classification used in machine learning. SVM's use what is called a kernel function to arrive at a threshold, called the decision boundary, which is compared against new data to provide a classification. There are three types of kernel functions that we will test: linear, polynomial and radial kernels.

ANN is a deep learning algorithm that is often used for classification problems. It learns the complex relationship between various features of the dataset using a 'Feed Forward' approach in which data is passed between nodes in a forward way only.

RF is an ensemble algorithm that is used for classification problems. It generates prediction models based on randomly selected features from the diversified decision trees.

The models will be trained on a time series dataset containing historical quantitative data such as price and indicator values which are used as features. Once the models are trained, they will be evaluated using five-fold cross validation. To measure the performance of the models, accuracy will be used. A prediction is considered accurate if the model prediction is in the same direction as the trend movement in the next day. Our goal is to have the models predict trend reversals for the next day using the closing price and indicator values of the correlated assets in the current day.

2 Main Results

With inverse assets, it is possible to predict the next day's price direction at a rate of 63% using SVM. Optuna and Random Search helped increase the accuracy of the models further by tuning their hyperparameters. However, significant improvement was made in the dataset by converting continuous indicator data to discrete values. These values were assigned based on certain overbought and oversold thresholds that are commonly used in trading. This approach increased the accuracy of models by a few more percent.

As the dataset was explored further, we realized predicting the direction of a trend yields a higher accuracy due to less noise. As a result, our focus changed to predicting the next day's trend of SPX using the inverse assets. Indicators were also customized further by equipping them with their own moving averages to further understand their individual movements and patterns. This strategy achieved an accuracy of 84% when trained on exclusively DXY, VIX, and US10Y. This is significant because it shows the correlation between the trends among the assets. The accuracy is also higher because a trend following strategy ignores random price movements of individual days that are in the opposite direction of the trend acting as noise.

3 Contributions

Most studies focus on predicting the price or direction of a single asset. However, this paper predicts the direction of an asset using inverse assets that move in the opposite direction. Opposite patterns can sometimes emerge faster among inverse assets than the main asset itself. The main dataset used for training was also created by merging and cleaning multiple datasets from each asset. The indicators (i.e., features) are also customized by equipping them with meaningful thresholds such as calculating the moving average of indicators to reduce their noise and better understand their movements with respect to their past week's average values.

4 Related Work

Most research work for analysing and predicting financial time series such as stocks mainly focus on Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Decision Tree (DT).

Work	Forecasting Model	Accuracy	Indicator
Patel et al. (2015)	ANN, SVM, Random Forest, Naive-Bayes classifiers	84%, 90%, 90%, 90%	Momentum, Stochastic K%, Stochastic D%, Relative Strength Index, Moving Average Convergence Divergence (MACD), etc
Manojlović et al. (2015)	Random Forest	76.5% (for 5 days), 80.8% (for 10 days)	5-days and 10 days moving average, Stochastic K%, Stochastic D%, MACD, Commodity channel index (CCI), etc
Kim (2003)	SVM	56%	%K, %D, Slow %D, Momentum, ROC, CCI, etc
Tsai el al. (2009)	ANN + DT, ANN, DT, DT + DT	77%, 59%, 65%, 67%	Fundamental and technical analysis
Polamuri et al. (2019)	Random Forest	61.9%	Fundamental and technical analysis

Table 1: Comparisons of related work.

Patel et al. (2015) states that they can predict stock prices by computing the trading data such as open, high, low, and close (OHLC) prices. For their study, they experimented with the data of two indices (CNX Nifty and S&P BSE Sensex), and two stocks (Reliance Industries and Infosys Ltd.) from Jan 2003 to Dec 2013. By normalizing these data, and applying ANN, SVM, Random Forest, and Naive-Bayes classifiers, the results predicted the future stock prices with the accuracy of nearly 84% for ANN, and 90% for the rest of the algorithms.

Similarly, Kim (Kim, 2003) applied SVM to predict the time series of the Korean stock market, and reported that the accuracy was around 56%. He further concluded that the SVM model outperformed the backpropagation neural network (BPN) and Case-based reasoning (CBR), by 3.0981% and 5.852% respectively, as SVM is more efficient in generalizing and minimizing the structural risks.

Manojlovič et al. (2015) experimented the 5-days-ahead and 10-days-ahead predictive models based on the CROBEX index and Zagreb Stock Exchange. They used the random forests algorithm with several technical indicators and resulted in 76.5% of accuracy for 5-days-ahead models and 80.8% for 10-days-ahead models. Although the predictions were efficient, they proposed that the model accuracy can be further improved by using some hybrid models.

Tsai et al. (2009) has created a stock price forecasting model based on the Taiwanese electrical stock market data by combining the Artificial Neural Networks (ANN) and Decision Tree (DT). Although they have found that ANN has an efficient predicting ability on stock market price compared to other models, they have identified that it cannot explain the cause and effect of the stock price change. Through the experiment using the sliding window technique, they confirmed that the combination of ANN and DT provided the explanation of the cause and effect in its forecasting ability. By comparing the performance of each model, DT + ANN reported to have a hit rate of 77%, which is higher than other models with 67% hit rate.

Polamuri et al. (2019) studied stock prediction based on Linear Regression, Multivariate Regression, Random Forest, and Extra Tree Regressor using the S&P 500 index. They have concluded that the Decision Tree and Random Forest Regressor were the best regression algorithm compared to other algorithms.

5 Methodology

The problem of predicting price direction is a classification problem because the outcome can be broken down into two classes, days where the asset rises in price and days where it falls. For this reason, we will focus on three classification models, Support Vector Machine, Artificial Neural Networks, and Random Forests. These models were chosen because they were the most commonly used models by previous researchers, with proven success. For instance, Patel (2015) was able to achieve 90% accuracy with both SVM and Random Forests.

5.1 Support Vector Machine (SVM)

We chose to use SVM because it is one of the more popular classification algorithms, and previous researchers have found success using it. SVM is a classification algorithm that uses a support vector classifier, along with "soft margins" to allow a small number of misclassifications Cortes (1995). This is useful since real world data is often linearly inseparable, and therefore there is no way to possibly classify the data without some misclassifications. Since our problem is a classification problem where we anticipate highly overlapping data, we expect SVM to perform well for our use case.

Additionally, we can use kernel functions to map the data into a higher dimension. Once in a higher dimension, we can more easily find a support vector classifier that most accurately classifies the data. In this project we will be using the linear, polynomial, and radial (RBF) kernel functions. We use these kernel functions to define relationships between data in a higher dimension. For example, if our data is the first dimensional, we can use a kernel function to translate the data into two dimensions where a line will serve as a better decision boundary than a single threshold would have in one dimension. The kernels are defined as Ali (2021):

$$Polynomial = (a \times b + r)^d \tag{1}$$

$$RBF = e^{-\gamma(a-b)^2} \tag{2}$$

where a and b are two different observations in the data set, r is the coefficient of the polynomial and d is the degree of the polynomial. In the radial function, γ is a constant factor. We will use cross validation to decide on values r,d, and γ. All of these kernels are already implemented by scikit-learn's machine learning library.

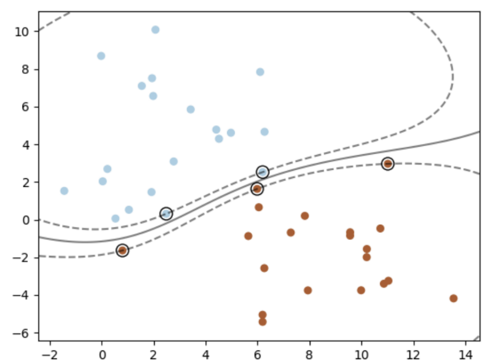


Figure 2: Example of Scikit-learn SVM using a radial kernel Pedregosa (2022).

5.2 Artifical Neural Nets (ANN)

ANN is a non-linear deep learning algorithm that can solve classification problems by learning the relationship between different features. ANN is made up of node layers such as the input, hidden, and output layers. Each input node receives one feature such as an indicator or the closing price of a security.

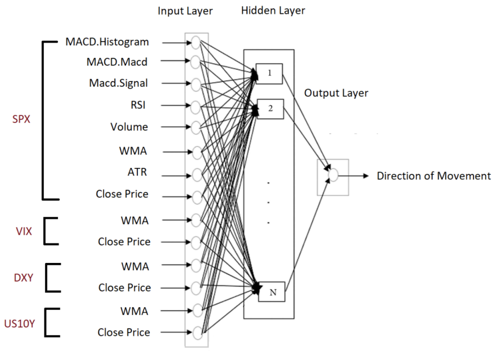


Figure 3: Image is modified to include indicators and information that we need Ali (2021).

The architecture above is showing a 'Feed-Forward' architecture in which data moves forward only. The weight and output of each node will be used by other nodes in the network Ali (2021). ANN accuracy is evaluated through its cost function, so the lower this value, the more accurate the model is. To optimize the model, we will be using 'Gradient Descent', an iterative optimization algorithm which can find the minimum value of the cost function by optimizing the weights and biases TradingView.

5.3 Random Forests (RF)

Random Forest consists of many decision trees which represent distinct instances of data. Decision trees start from the root and split its node based on the features as a binary tree. The classifier determines the outcome at each node is True or False, and travels down the tree accordingly until it reaches the leaf node Ho. (1995). The overall outcome is only determined once the leaf node is reached, without being pruned in the middle, thus providing a more accurate result.

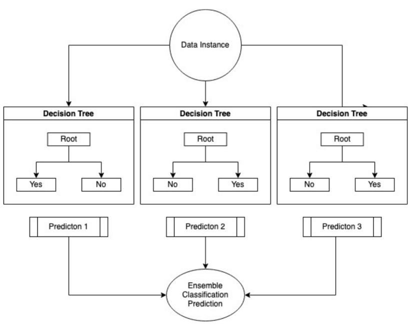


Figure 4: Ensemble classification prediction by random forest using multiple decision trees.

With the outcomes from each decision tree, Random Forest utilizes ensemble learning and bootstrapping to make the final classification Education.. Ensemble learning refers to processing multiple models and averaging out the results of each model. Since we are using random forests for classification, bootstrapping consists of a majority vote, meaning the classification that more decision trees arrive at will be the overall classification. With the combination of two, random forest generates a strong prediction classification result.

In this paper we are testing our data using black box models. The black box model does not reveal the inner workings for its data. The downside to this is that we cannot visualize how a model comes to its conclusion. Black boxes can be used to protect the proprietary software. For the purposes of this paper, we will go not into the depth of the inner workings of the machine learning algorithm. Rather, the test results from using the algorithms are more important as we want to test how well our pre-processed data has been optimized.

5.4 Accuracy and Parameter Tuning

The models are evaluated by comparing their accuracy on the test data. This paper focuses on Accuracy over precision and recall because the goal is to maximize the number of correct predictions, both up and down. Precision on the other hand would only focus on how well our models predict in one direction, and the same is true for recall.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

The algorithms behind our models were excellent at predicting when the price will rise but terrible at predicting when the price will fall. One of the main reasons is that, bear markets (i.e., a down trending market), tend to last a short time because investors sell their assets quickly due to fear of losing more capital. Once the markets are oversold, institutions jump in to grab those under-priced assets which ends up moving the price up again, therefore, bear markets are rare and quick. We believe our models have not seen enough samples of these time periods to accurately predict, and warn us about it.

To compare the models more accurately, the dataset was randomly split in five ways using a five-fold cross validation which helped prevent overfitting. Scikit-learn library was also used to further optimize the models by finding the parameters to train them with. Scikit-learn has a function called RandomizedSearchCV that randomly generates combinations of hyper parameters and yields the best solution using cross validation. Although one may argue that no intelligence is used in selection of the parameters, it can find the optimal hyperparameter when a ‘uniform’ search, such as in grid search, may hop over the optima parameter.

In addition, Optuna is also used for hyperparamter tuning because it is lightweight and versatile. It uses a familiar pythonic syntax and quickly searches large spaces while also pruning any unwanted parameters. This framework dynamically generates the search space, and it provide various algorithms to search with. Here, we are using random search to compare its output with the accuracy of random search from the scikit-learn library.

5.5 Dataset

The dataset is collected from TradingView on the daily time frame for each security listed below. Then all datasets are merged into one main dataset based on the ‘date’ column.

Security	Market	Description
SPX	Equity	S&P 500 Index tracks the stock performance of 500 U.S. large companies.
US10Y	Fixed Income	U.S. 10 Years Treasury Bond Yield indicates the interest rate that government bonds are paying over a ten years maturity period.
DXY	Foreign Exchange	U.S. Dollar Index tracks the international value of the U.S. dollar currency.
VIX	Chicago Board Options Exchange (CBOE)	Volatility Index measuring the volatility of S&P 500 index options.

Table 2: The following financial securities will be used to train and test our models.

SPX is the index that we would like to predict the direction for, the other symbols are included to help us in making better predictions. US10Y, DXY, and VIX performance are all inversely correlated with equities. When equities show weakness, the fear in the market causes VIX to rise up. As investors start liquidating their assets, DXY goes up because investors are holding more cash. Based on the macroeconomic landscape of the market, investors could also sell off bonds which will decrease bond prices and increase the bond yield. That is why the combination of these datasets can work great in predicting the direction of the equity market, SPX.

However, there are a few limitations in our dataset. The main problem is that patterns are not consistent. Based on the macroeconomic situation, patterns change, so during a market crash such as 2020, investors purchased government bonds because it was guaranteed to be paid out causing the bond yield to go down in order to control demand. However, in the crash of 2022 due to high inflation, investors actually sold their government bonds because of the increase in interest rates by the U.S. Federal Reserve System. The reason is the increase in interest rates causes the purchased bonds to be worth less as the interest rates go up, so it is better for investors to wait until rates are halted or go down. We can see that both markets experienced a crash, but they were different in nature, and as a result, the former patterns died out and new ones emerged.

Another limitation we have is that bear markets (i.e., down trending markets) tend to last a short time because investors tend to sell their positions quickly due to fear of further capital loss. This means there is not enough sample data in the dataset to train the models with in order to understand and predict market crashes earlier. Most of the dataset consists of upward trends that last a long time. To fix this issue, we decided to look at data in lower time frames with the hope of having a bigger sample size, but it introduced more noise and randomness which decreased the accuracy of the models.

We also noticed that there is some degree of randomness in the data which causes some days to go in the opposite direction of the trend. It could be because of a certain economic reports or new company earnings which can cause massive move to the upside or downside in the span of a single or several days. These days mostly act as noise, so it is best to ignore them and instead follow the trend as long as it lasts.

5.6 Features

Indicator	Purpose	Formula
WMA Weighted Moving Average	A moving average that weighs the price of the most recent data more than the old data.	$WMA = \frac{(Price_1 \times n) + (Price_2 \times (n-1)) + ...Price_n}{\frac{n \times (n+1)}{2}}$, $n = last\ 21\ days$
MACD Moving Average Convergence Divergence	An indicator that works by the crossing of two exponential moving averages. It helps with identifying trends.	$MACD = fast.ema - slow.ema$, $slow.ema = last\ 26\ days\ EMA$, $fast.ema = last\ 12\ days\ EMA$
EMA Exponential Moving Average	A moving average that places a greater weight and significance on the most recent data points.	$EMA = (Price \times (\frac{smoothing}{1 + days}) + ema_{yesterday} \times (1 - (\frac{smoothing}{1 + days}))$ $n = last\ 9\ days$
RSI Relative Strength Index	An oscillator that measures the overbought and oversold condition of a security, and as well as momentum.	$RSI = 100 - \frac{100}{1 + (avg.gain/avg.loss)}$ $avg.gain = \frac{average\ gain}{n}$, $avg.loss = \frac{average\ loss}{n}$, $n = last\ 14\ days$
ATR Average True Range	It measures the overbought, oversold, and the momentum of an asset.	$TR = Max[(High - Low), Abs(High - Close), Abs(Low - Close)]$, $ATR = (\frac{1}{n}) \sum_{i=1}^{n} TR_i$, $TR_i = a\ particular\ true\ range$, $n = time\ period$

Table 3: Feature definitions and their formula's

The datasets contain indicator values which be used as features for the models. On TradingView, there are certain smoothing techniques applied to features on top of their underlying math formula to remove noise. We decided it is best to let the platform handle that for us because it will ensure the data we work with is what we see on the charts. These features are quantitative in nature because they have a numerical value, and this is based on a math formula which varies per indicator. When an indicator reaches a certain value, it usually indicates a buy or sell signal. That is why we can use them to identify bullish or bearish patterns. Let us consider the RSI indicator as an example, if the value of RSI is at 30 or less, the underlying asset is considered oversold which means it is time to buy. However, if the value is 70 or more, the asset is considered overbought which means it is time to sell such as the example below.

Another feature is MACD, which works based on the cross of two moving averages. When the histogram turns positive, it means there is an upward trend, and if it turns negative it means there is a downward trend. We have also equipped these features with their own moving averages by calculating their 7 days moving average to better understand the change in the indicators themselves. All these features will be used for every asset in order to achieve the highest accuracy in the models.

6 Results

6.1 System Design

As shown in Figure 5, most of the data was obtained from TradingView, where we then converted the data into discrete values and assigned them labels. Initially, the spx.result (i.e., the feature we were going to predict) was calculated based on the next day's close of SPX because we were going to predict its next day's direction. However, this approach yielded a lower accuracy because it had a lot of noise due to the random price movements we talked about. We then decided to calculate the spx.result column based on the SPX trend which meant we would ignore the days that would go against the main trend acting as noise. We would only change our trade from long to short or the other way around when the trend would reverse.

After the data was labelled, we then passed the data onto our parameter optimizing methods to find the best parameters to train our models with. The best parameters are not included in this paper because we believe they are subjective because if we use a bigger or smaller training data, the values of the optimized parameters would change, so including them would not have added any value. To train our models, the data was split randomly into 80% training data and 20% testing data. The training data was then combined with the parameters from our parameter optimization, and used to fit the model. At last, the models were used to classify the test data, and we analysed the result to find the accuracy.

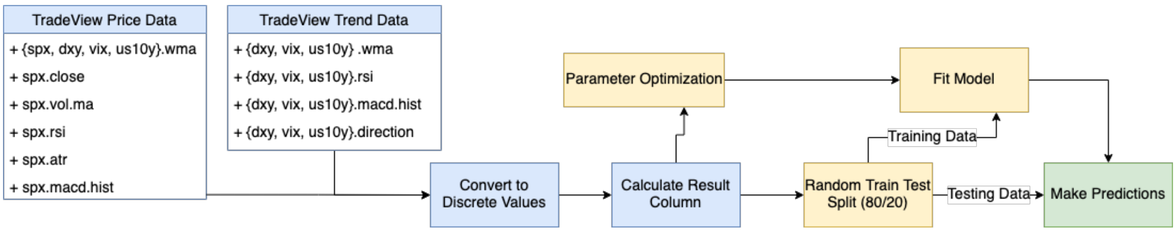


Figure 5: Flow of data through our models

6.2 Sigle Day Direction Strategy

The initial goal was to accurately predict the price direction of the S&P 500 for the next day using inversely correlated assets. As seen in Table 4, the models initial implementation had some success, but there was room for improvement. The first strategy was to label the dataset. Similar to Patel (2015), the indicator values were converted from their continuous form to discrete values based on whether they were considered bullish or bearish patterns. This improved our results to some extent. For example SVM model went from 54.9% accuracy to 57.7%. In addition to labelling our data, we tried using data from a lower time frame in order to obtain more sample data to train and test the models with. We tried as frequent as 10 minutes, but this approach resulted in a lower accuracy due to the more noise and randomness in the price movement of assets.

6.2.1 Hyper-parameter Tuning

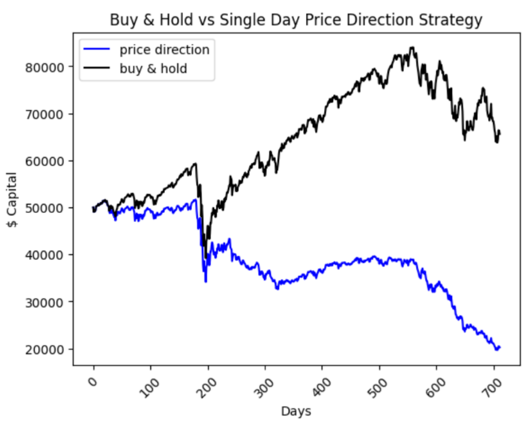
Another strategy we used was to tune the hyperparameters of models which improved the accuracy to some extent. Both Random Search and Optuna was used to test the models. Between the two, Optuna performed better for both SVM and RF.

Model	No Tuning	Random Search	Optuna
SVM	57.7%	56.9%	62.6%
ANN	55.0%	56.3%	55.9
RF	55.3%	51.1%	56.5%

Table 4: Model performance for predicting the price direction of SPX on individual days

6.2.2 Backtesting

In table 4, SVM performed the best out of the three models with an overall accuracy of 62.6%. From 2019 to 2022, the model executed around 126 trades in total; both long and short. If we had given it $50,000, it would have have lost money and ended up with $20,230 in total which shows this strategy is not profitable.



6.3 Trend Following Strategy

As exploratory research, we began to investigate how accurately we could predict the next day trend of SPX. As it turns out, the models could predict next day trends with an accuracy of 90% using SPX data included. This was very exciting as trading based on trends is valuable in swing trading. However, we discovered that the same results could be obtained by simply predicting tomorrow's trend will be the same as today's trend. On top of that, of the 258 changes in direction in our testing data, SVM was only able to predict the change exactly one day before on 9 occasions. This led us to investigate further into predicting the trend of SPX based on the inverse assets alone. We found that SVM was able to obtain an accuracy of 84% even with no data from SPX. While this result is clearly not as good as the 90% we got with the SPX information included, it clearly shows the correlation between the inverse assets and SPX data.

6.3.1 Hyper-parameter Tuning

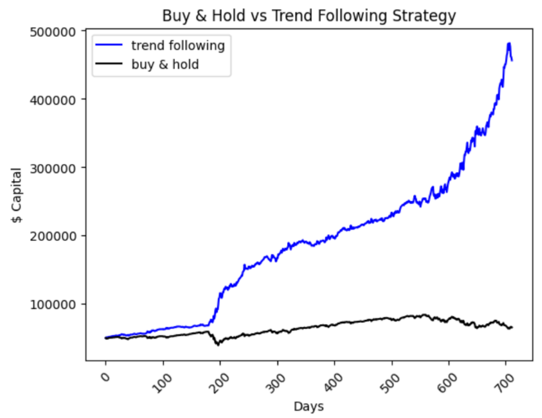
Based on table 5, we can see that hyperparameter tuning increased the accuracy of our models by suggesting better parameters. Random Search performed better than Optuna for all the models, and ANN yielded the highest accuracy among all other models. We believe that Random Search works best when there is less noise in the train-test data, and a trending strategy exactly provides that.

Model	No Tuning	Random Search	Optuna
SVM	77.9%	83.8%	80.5%
ANN	79.8%	81.2%	80.6%
RF	81.7%	82.0%	80.1%

Table 5: Models performance for predicting the trend direction of SPX

6.3.2 Backtesting

Based on table 5, again SVM performed the best out the three models with an overall accuracy of 83.8%. From 2019 to 2022, the model executed the total of 258 trades (i.e., both long and short trades). With $50,000 capital to begin with, the model would end up with a total of $456,307.



This backtesting strategy assumes the profit or loss is compounded at the end of each trade without taking taxation, commission, and other fees into account. When the trend is predicted to be up, we assume a long position, and when it is predicted to shift down, we move to a short position.

7 Discussion

7.1 Interpretation

When we started our research, we were wondering if using inversely correlated assets would help us predict the next day direction or overall trend of the S&P 500. Using SVM, we have been able to predict whether or not the price of S&P 500 will go up tomorrow with an accuracy of 63%. With SVM, we can predict the next day trend with an accuracy of 84% without any data from SPX at all. Since the accuracy only dropped from 90% to 84% when removing all SPX data, we can be sure that there must be a correlation between SPX, and the inverse assets.

These results indicate that while the overall accuracy of our models is not higher than all previous work, inversely correlated assets are helpful in predicting SPX trend. The fact that we were ultimately able to predict trend direction solely using the US10Y, DXY, and VIX was exciting and good achievement.

7.2 Implications

The results of our research are similar to that the findings of previous researchers. Our price direction prediction of 63% is higher than that of Kim (2003), and Polamuri* (2019), however it is surpassed by other research such as Patel (2015). These researchers use technical analysis on the individual assets whereas we used multiple assets to train our models with. This introduces a level of uncertainty that they would not have faced when it comes to correlation between the features and the result. In the financial industry, it is more common for each asset class or sector to have their own model. Perhaps, in the future we could try to assign each inverse asset their own model, so we can predict their own trend, and then once we have their individual predictions, we could then predict the SPX trend direction for the next day.

To achieve profitability, we do not need to find a strategy with +90% accuracy, but instead we need to find strategies that are reliable and consistent in terms of recurring patterns. The less obvious these patterns are, the more rare and profitable they could be because they have not been exploited by other competitors. We can see that even with a well-known trend following strategy, it is still possible to achieve profitability.

7.3 Limitations

While we have shown that there are correlations between the inverse assets and SPX, predicting the exact day of a change in trend was not very successful. As previously mentioned, SVM was only able to accurately predict the day of a change in trend 9 out of 258 times. We have to admit that our model has not seen enough samples of down trending markets because they occur and disappear quickly compared to bull markets that uptrend for a long time.

A trend following strategy works best with a small capital because huge amounts of capital cause market slippage which causes the price of the asset to shift up or down significantly making it difficult for the trader to enter or exit at the desired price. There has to also be enough shares to purchase or enough buyers to want to purchase our shares at a given time, so this strategy only works in liquid markets such as stocks with a large market cap or ETFs.

8 Conclusion

Due to the lucrative nature of the financial market, lots of research has been done to correctly predict stock prices. The stock market however is highly volatile and it is influenced by many variables such as unexpected major events and uncertainty in politics and the economy. It has become popular to incorporate machine learning that takes these uncertainties into account and use indicators and patterns to predict stock trends accurately.

Our pre-processing allowed us to find correlations within the SPX and inverse asset's data for better classification of SPX's trend. The predictions made by our models with hyperparameter tuning resulted in an increase in accuracy over the models trained without the tuning. This indicates that the pre-processing step was as efficient as hyperparameter tuning.

Recurrent Neural Networks (RNNs) is one of the popular models used to reduce the prediction error by obtaining the lost values during the data training. A possible future direction of our research could be using RNN to reduce error. The Long Short-term Memory (LSTM) model could also be tested as it is one of the powerful models that can capture the patterns and predict trends.

In our paper, we focused on the overall accuracy of our predictions. A possible area for future exploration could be focusing on precisely predicting the changing of trends. For example, the algorithm could provide the exact date when the trend ends, or even the exact time.

It would be interesting to try training a model using the inverse assets on a distinct times of similar macroeconomic climate. For instance, training a model on data from the financial crisis of 2008, and testing it on a market crash such as the market crash of March 2020. As we have shown, models trained on inversely correlated assets can spot trends among other assets such as SPX.

References

Muhammad Aamir Amjad Ali Zubair Ahmad Ali, Dost Muhammad Khan. Predicting the direction movement of financial time series using artificial neural network and support vector machine. *Hindawi*, 2021. URL https://doi.org/10.1155/2021/2906463.

Y.-W. Chang. Training and testing low-degree polynomial data mappings via linear svm. *Journal of Machine Learning Research 11*, 2010.

Shah M. Kshirsagar A. Chhajer, P. The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytics Journal, 2, 100015.*, 2022. DOI: https://doi.org/10.1016/j.dajour.2021.100015.

Vapnik V. Cortes, C. Support vector networks. 1995. DOI: https://doi.org/10.1007/BF00994018.

IBM Cloud Education. What is random forest? URL https://www.ibm.com/cloud/learn/random-forest.

Tin Kam Ho. Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition.*, 1995. DOI: https://doi.org/10.1109/icdar.1995.598994.

K.-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 2003. DOI: https://doi.org/10.1016/s0925-2312(03)00372-2.

H. Jabani S. S. M. Nabipour, P. Nayyeri and A. Mosavi. Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. *IEEE Access*, 2020. DOI: https://doi.org/10.1109/ACCESS.2020.3015966.

T. Manojlović and I. Štajduhar. Predicting stock market trends using random forests: A sample of the zagreb stock exchange. *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015. DOI: doi: 10.1109/MIPRO.2015.7160456.

C. F. A. Margaret Yang. Sp 500 may be vulnerable to a pullback if the us dollar rebounds. URL https://www.dailyfx.com/forex/fundamental/forecast/weekly/title/2020/12/06/SP-500-May-Be-Vulnerable-To-a-Pullback-if-the-US-Dollar-Rebounds.html.

Shah S. Thakkar P. Kotecha K. Patel, J. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 2015. DOI: https://doi.org/10.1016/j.eswa.2014.07.040.

Varoquaux G. Gramfort A. Michel V. Thirion B. Grisel O. Blondel M. Prettenhofer P. Weiss R. Dubourg V. Vanderplas J. Passos A. Cournapeau D. Brucher M. Perrot M. Duchesnay É. Pedregosa, F. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research.*, 2022.

Srinivasi D. K. Mohan D. A. Polamuri*, S. R. Stock market prices prediction using random forest and extra tree regression. *International Journal of Recent Technology and Engineering (IJRTE)*, 2019. DOI: https://doi.org/10.35940/ijrte.c4314.098319.

E. Su. Capital markets volatility and covid-19: Background and policy responses. URL https://crsreports.congress.gov/product/pdf/R/R46424.

TradingView. Live stock, index, futures, forex and bitcoin charts on tradingview. URL https://www.tradingview.com/chart.

N. Tripathy. Stock price prediction using support vector machine approach. *Proceedings of The International Academic Conference on Management and Economics.*, 2019. DOI: https://doi.org/10.33422/conferenceme.2019.11.641.

Wang. Tsai, Chih-Fong P. *Stock Price Forecasting by Hybrid Machine Learning Techniques.*, chapter 1. Lecture Notes in Engineering and Computer Science., 2009.

Yale. United states stock market confidence indices. URL https://som.yale.edu/centers/international-center-for-finance/data/stock-market-confidence-indices/united-states.