# A Machine Learning Long-Short Approach to Market Timing and Trend Following

Adnane Deroui
*Master in Data Science and Business Analytics*
*International University of Rabat*
Rabat, Morocco

***Abstract*—In this study, we developed an algorithmic trading strategy utilizing a Deep Neural Network (DNN) to predict market trends and optimize portfolio returns. The strategy was rigorously backtested against historical market data, comparing it to a traditional buy-and-hold strategy. To assess performance, we analyzed key metrics such as cumulative returns, Sharpe ratios, and drawdowns. Additionally, Monte Carlo simulations were employed to generate 3,000 random paths based on Brownian motion, forecasting the next 252 trading days. The results highlighted the strategy's robustness, with the simulated paths showing a median performance of 40% and moderate drawdowns, further supporting the strategy's effectiveness in varying market conditions. The mean maximum drawdown across simulations was 12.5%, indicating controlled risk, while the DNN model achieved strong cumulative returns and favorable risk-adjusted performance.**

## I. Introduction

Financial markets are inherently volatile, with prices fluctuating based on a wide range of factors. Timing market entries and exits is a complex challenge that can significantly impact the performance of trading strategies. Traditional methods rely heavily on fundamental and technical analysis, but recent advancements in machine learning offer new approaches to optimizing market timing. A *long-short market timing strategy* involves taking long positions in assets expected to appreciate and short positions in those expected to decline, offering a balanced approach to risk management.

In this study, we propose a machine learning-based long-short market timing strategy. We will utilize various technical indicators, such as the *Moving Average Convergence Divergence (MACD), Moving Averages*, and the *Rate of Change (ROC)*, to capture market trends and momentum. To enhance the efficiency of our model, we will apply *Principal Component Analysis (PCA)* for dimensionality reduction, ensuring that the most relevant information from the technical indicators is retained while reducing noise and overfitting.

A key component of our strategy's robustness lies in the application of Monte Carlo simulations. These simulations, incorporating Brownian motion, will allow us to assess the performance of our model under various market conditions by generating numerous potential price paths. This testing will ensure that our strategy is not only effective in historical data but also adaptable to future market fluctuations.

## II. Literature Review

### A. Market Timing and Trend Following:

Market timing involves predicting future market movements to determine when to buy or sell assets. Accurate market timing can lead to higher returns and reduced risk, although it is challenging due to the unpredictable nature of financial markets. Traditional methods of market timing primarily relied on economic and fundamental analysis [1]. However, with advancements in computational finance, modern approaches now incorporate data-driven strategies, including machine learning, to improve prediction accuracy.

Trend following is a related strategy that focuses on identifying and acting upon the direction of asset prices. Instead of predicting price reversals, trend followers aim to enter the market when a trend is confirmed and exit when signs of trend exhaustion appear. This method capitalizes on market momentum, which suggests that assets that have performed well in the past will continue to do so in the near future [2]. Jegadeesh and Titman first documented the momentum effect, showing that it persists across various markets and time frames [3]. Machine learning techniques have further enhanced the effectiveness of trend-following strategies, as Patel et al. (2015) demonstrated that combining machine learning algorithms with momentum-based indicators improved short-term price movement predictions [4].

### B. Technical Indicators:

Technical indicators are mathematical formulas applied to price, volume, and open interest data to predict future price movements. These indicators are often categorized as momentum, trend, volatility, and volume indicators.

Momentum indicators, like the Moving Average Convergence Divergence (MACD) and Rate of Change (ROC), help traders evaluate the speed of price changes. MACD, introduced by Appel [5], uses two moving averages to identify momentum shifts and possible trend reversals. Similarly, ROC compares current prices to those from a previous period, providing insights into price velocity.

$$MACD = EMA_{12}(P) - EMA_{26}(P)$$

$$EMA_{12(P)} : 12\ days\ exponential\ moving\ average$$
$$EMA_{26\,(P)} : 26\ days\ exponential\ moving\ average$$

The Simple Moving Average (SMA) is one of the most widely used trend indicators. It calculates the average price of an asset over a specific period, smoothing out short-term fluctuations to identify the direction of a trend. The SMA is popular due to its simplicity and effectiveness in tracking long-term market trends.

$$SMA_n = \frac{\sum_{i=1}^{n} P_i}{n}$$


Fig. 1. Moving Average Indicating Trend

Volatility indicators, such as Bollinger Bands and the Average True Range (ATR), measure market volatility by showing the price variation over a given time frame. Bollinger Bands, developed by John Bollinger [6], indicate potential price breakouts, while ATR provides insight into price volatility by averaging the true range over a set period.


Fig. 2. Upper and Lower Bollinger Bands

Volume indicators, such as *Volume* and *Relative Volume*, are crucial in assessing the strength of market movements. Volume represents the number of shares or contracts traded during a specific period and is often used to confirm the validity of price trends. A significant increase in volume typically signals strong market interest, suggesting that the current trend may continue.

Relative Volume compares the current volume to the average volume over a specified period, highlighting unusual trading activity. A higher RVOL indicates increased market participation, which can signal a potential breakout or trend reversal, making it a valuable tool for traders to assess the strength behind price movements.

Fig. 3. High Volumes Indicating Strong Price Movement

## C. *Monte Carlo Simulations*

Monte Carlo simulations are a key tool in assessing the robustness of financial models and strategies. They use random sampling from probability distributions to simulate thousands of possible future price paths, typically assuming that asset prices follow a *Brownian motion process* [9]. Originally developed by Metropolis and Ulam, Monte Carlo simulations have become essential in risk management and derivatives pricing.

$$S_t = S_0 e^{\sigma W_t \frac{-1}{2}\sigma^2 t}$$

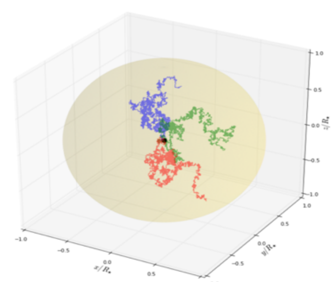

Fig. 4. Representation of the Brownian Random Walks in 3D

For market timing strategies, Monte Carlo simulations allow traders to test models under various hypothetical scenarios, ensuring that strategies are robust across different market conditions. Pardo (2008) explained that stress testing a trading system using Monte Carlo simulations can uncover weaknesses that might not appear in traditional backtesting [10]. By simulating multiple price paths, traders gain a deeper understanding of potential risks and variability in returns.

Monte Carlo simulations also play a critical role in risk-adjusted performance evaluations. When combined with machine learning-based strategies, these simulations provide insights into how factors such as market volatility and regime changes impact overall performance.

### *D. Machine Learning Used in Quantitative Finance*

- Support Vector Machine (SVM):
  SVM is effective in both regression and classification tasks. It finds the optimal boundary separating classes by transforming data into a higher-dimensional space through the kernel trick. SVM has been applied in stock market prediction to classify market trends, as demonstrated by Indu Kumar et al. and Wen Fenghua et al., who also used Singular Spectrum Analysis (SSA) to further improve the model's accuracy.

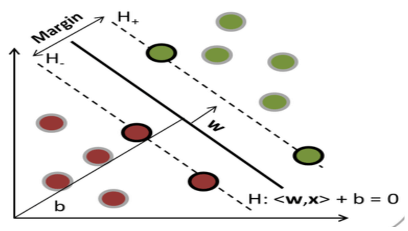


Fig. 5. Representation of Support Vector Machine

- Random Forest (RF):
  Random Forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy. It handles large datasets efficiently and reduces overfitting. Studies such as those by Indu Kumar et al. and Mehar Vijh et al. have shown that RF can outperform traditional decision tree algorithms in stock market forecasting.

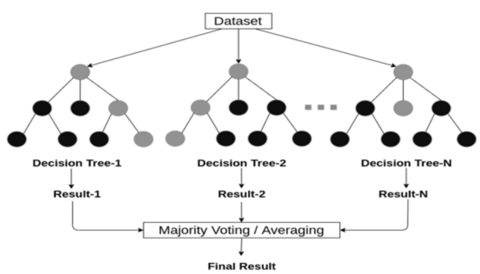


Fig. 6. Random Forest Algorithm with multiple Decision Trees

- K-Nearest Neighbors (KNN):

KNN is a non-parametric method that classifies data points based on the closest labeled examples. It has proven useful in stock market prediction by grouping similar historical data points to estimate probabilities of future price movements, as evidenced by research from Indu Kumar et al.

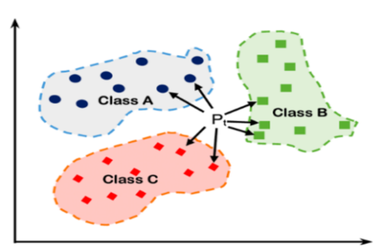


Fig. 7. K-Nearest Neighbors Algorithm

- Naïve Bayes (NB):

Naïve Bayes is a probabilistic classifier that predicts the likelihood of a class based on prior knowledge. NB is particularly suited for real-time stock market prediction due to its fast-learning process. Studies like those by Indu Kumar et al. demonstrate the algorithm's ability to forecast market behavior by leveraging historical data.

$$P(A/B) = \frac{P(B/A) \times P(A)}{P(B)}$$

P(A/B) : probability of event A given B
P(B/A) : probability of event B given A
P(B) : probability of event B
P(A) : probability of event A

- Artificial Neural Network (ANN):

ANNs mimic the human brain's neural structure and are used to learn complex patterns. In stock market prediction, ANNs detect trends in historical price data and make accurate forecasts. Unlike linear models, ANNs use hidden layers to process information and improve accuracy, as shown in research by Wen Fenghua et al.

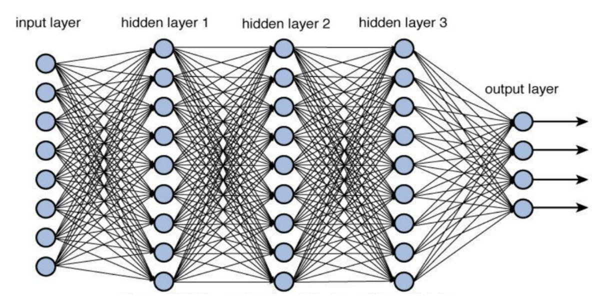


Fig. 8. Deep Neural Network with Multiple Layers

## III. METHODOLOGY

### *A. Problem Statement*

The primary problem this study aims to address is the development of an effective long-short market timing strategy using machine learning algorithms to optimize buy and sell decisions in financial markets. Traditional market timing methods often rely on subjective judgment or static rules, which may be prone to error and inefficiency in rapidly changing market conditions. With the increasing complexity and volatility of financial markets, there is a need for a more robust, data-driven approach that can adapt to dynamic trends and minimize risks. Without accurate market timing techniques, investors may miss profitable opportunities or incur unnecessary losses, impacting overall portfolio performance. This study seeks to overcome these challenges by leveraging machine learning models, technical indicators, and advanced simulation methods to enhance the precision and reliability of market timing strategies.

## *B. Importing Libraries*

Python offers a wide range of libraries that make data analysis, visualization, and machine learning more accessible and efficient. Below are the libraries used in this project

| | |
|---|---|
| ***Pandas*** | *Python library for data manipulation and analysis* |
| ***Numpy*** | *Python library for linear algebra using arrays and matrices* |
| ***Matplotlib*** | *Python library for basic data visualizations* |
| ***Seaborn*** | *Python library for advanced data visualizations* |
| ***Scipy*** | *Python library for scientific and technical computing* |
| ***Scikit-learn*** | *Python library for machine learning algorithms* |

Table 1. Libraries Used for this project

## *C. Overview of the Breast Histopathology Dataset*

The dataset for this study consists of historical stock market data collected from Yahoo Finance, a widely recognized and reliable financial data provider. This platform offers comprehensive and accurate data, making it a trusted source for financial analysis. The dataset includes daily price information such as open, high, low, and close prices, along with trading volume, which is crucial for calculating technical indicators. Additionally, we will test our strategy with US stocks, specifically the SPY index, which tracks the performance of the S&P 500, one of the most significant and liquid stock market indices globally. By using this data, we ensure the robustness and relevance of the results for real-world market conditions.

| Date | Open | High | Low | Close | Adj Close | Volume | Returns_1d | Returns_5d | Returns_20d | Returns_10d | ... | BBL_100 | BBW_100 | BBH_200 | BBL_200 | BBW_200 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1994-01-27 | 47.406250 | 47.812500 | 47.343750 | 47.750000 | 27.499775 | 344500 | 0.009247 | 0.005925 | 0.019346 | 0.011251 | ... | 45.595782 | 1.944061 | 47.737999 | 43.802314 | 3.935685 |
| 1994-01-28 | 47.937500 | 48.031250 | 47.875000 | 47.875000 | 27.571772 | 356500 | 0.002618 | 0.010554 | 0.027498 | 0.009888 | ... | 45.590187 | 1.994627 | 47.771463 | 43.798225 | 3.973238 |
| 1994-01-31 | 48.062500 | 48.312500 | 48.000000 | 48.218750 | 27.769747 | 313800 | 0.007180 | 0.021854 | 0.037660 | 0.017139 | ... | 45.567472 | 2.084431 | 47.812907 | 43.791468 | 4.021439 |
| 1994-02-01 | 48.156250 | 48.156250 | 47.906250 | 47.968750 | 27.625763 | 303600 | -0.005185 | 0.016556 | 0.028131 | 0.010533 | ... | 45.549123 | 2.152378 | 47.845131 | 43.793619 | 4.051513 |
| 1994-02-02 | 48.125000 | 48.281250 | 48.093750 | 48.281250 | 27.805731 | 307600 | 0.006515 | 0.020476 | 0.032754 | 0.019802 | ... | 45.518505 | 2.250491 | 47.885007 | 43.791555 | 4.093452 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2024-09-11 | 548.700012 | 555.359985 | 539.960022 | 554.419983 | 554.419983 | 75248600 | 0.010259 | 0.006298 | 0.022840 | -0.012715 | ... | 503.085298 | 69.902204 | 575.473109 | 454.312691 | 121.160418 |
| 2024-09-12 | 555.010010 | 559.400024 | 552.739990 | 559.090027 | 559.090027 | 51892700 | 0.008423 | 0.017249 | 0.028212 | 0.001415 | ... | 504.564715 | 68.221970 | 575.717405 | 455.106296 | 120.611109 |
| 2024-09-13 | 559.710022 | 563.030029 | 559.450012 | 562.010010 | 562.010010 | 39310500 | 0.005223 | 0.040066 | 0.016164 | 0.006555 | ... | 505.792239 | 67.012721 | 575.987962 | 455.911038 | 120.076924 |
| 2024-09-16 | 561.739990 | 563.109985 | 559.900024 | 562.840027 | 562.840027 | 36656100 | 0.001477 | 0.030069 | 0.015389 | -0.001490 | ... | 506.734418 | 66.272165 | 576.261956 | 456.716145 | 119.545810 |

Fig. 7. Overview of our Dataset

## *D. Exploratory Data Analysis*

In this section, we perform an Exploratory Data Analysis (EDA) to better understand the structure and key characteristics of the dataset. EDA is essential for identifying trends, spotting anomalies, and providing a general overview of the data before diving into any deep learning algorithms.

### *1) Statistical Summary*

| Date | Open | High | Low | Close | Adj Close | Volume | Returns_1d | Returns_5d | Returns_20d | Returns_10d |
|---|---|---|---|---|---|---|---|---|---|---|
| 1994-01-27 | 47.406250 | 47.812500 | 47.343750 | 47.750000 | 27.499769 | 344500 | 0.009247 | 0.005925 | 0.019346 | 0.011251 |
| 1994-01-28 | 47.937500 | 48.031250 | 47.875000 | 47.875000 | 27.571766 | 356500 | 0.002618 | 0.010554 | 0.027498 | 0.009888 |
| 1994-01-31 | 48.062500 | 48.312500 | 48.000000 | 48.218750 | 27.769735 | 313800 | 0.007180 | 0.021854 | 0.037660 | 0.017139 |
| 1994-02-01 | 48.156250 | 48.156250 | 47.906250 | 47.968750 | 27.625746 | 303600 | -0.005185 | 0.016556 | 0.028131 | 0.010533 |
| 1994-02-02 | 48.125000 | 48.281250 | 48.093750 | 48.281250 | 27.805731 | 307600 | 0.006515 | 0.020476 | 0.032754 | 0.019802 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2024-09-12 | 555.010010 | 559.400024 | 552.739990 | 559.090027 | 559.090027 | 51892700 | 0.008423 | 0.017249 | 0.028212 | 0.001415 |
| 2024-09-13 | 559.710022 | 563.030029 | 559.450012 | 562.010010 | 562.010010 | 39310500 | 0.005223 | 0.040066 | 0.016164 | 0.006555 |
| 2024-09-16 | 561.739990 | 563.109985 | 559.900024 | 562.840027 | 562.840027 | 36656100 | 0.001477 | 0.030069 | 0.015389 | -0.001490 |
| 2024-09-17 | 565.099976 | 566.580017 | 560.789978 | 563.070007 | 563.070007 | 49321000 | 0.000409 | 0.026021 | 0.006183 | 0.019907 |
| 2024-09-18 | 563.739990 | 568.690002 | 560.830017 | 561.400024 | 561.400024 | 59044900 | -0.002966 | 0.012590 | 0.004833 | 0.018967 |

Fig. 7. Statistical Summary

The dataset consists of 7,714 observations with considerable variability in stock prices and trading volume. The stock prices have a mean around 183 with a wide range from 43 to 565, indicating significant price fluctuations. Trading volume varies greatly, with an average of 86.69 million and a maximum exceeding 871 million, reflecting both quiet and highly active trading days.

Return variables show small average gains, but with occasional extreme price movements, evidenced by a wide range of returns (from -31% to 23%). The data spans from 1994 to 2024, and the target variables are balanced for predicting positive and negative outcomes.

### *2) Average Returns by Year*

The average daily returns by year provide valuable insights into market performance over time. Notably, 1995, 1997, 1998, and 2013 stand out with strong positive average returns, reflecting periods of market growth and favorable conditions. Years like 2000, 2001, 2002, and especially 2008 experienced negative average daily returns, highlighting challenging economic times such as the dot-com bubble burst and the global financial crisis of 2008, where returns significantly declined.

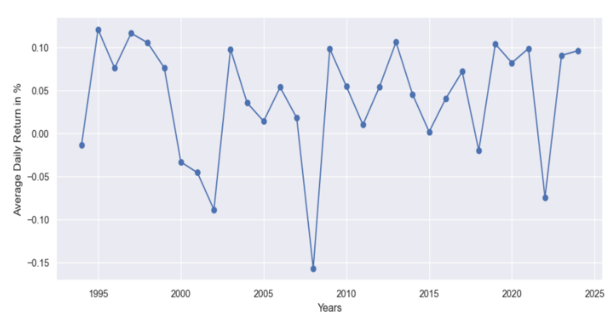


Fig. 8. Average daily return evolution year by year

More recent years, such as 2019, 2020, and 2021, show consistent positive returns, indicating market recovery and strength, likely driven by post-recession growth and stimulus efforts. However, 2022 marked a downturn with negative returns, reflecting potential market corrections or economic uncertainties during that period. The gradual recovery in 2023 and 2024, with returns nearing 0.001, suggests a return to more positive performance, potentially due to favorable market conditions post-pandemic or other economic recovery factors.

#### 3) Average Return by Month

The analysis of average daily returns by month reveals notable seasonal trends in market performance. November, April, and October show the strongest returns, while September has the worst performance, supporting the "September Effect." Summer months like June and August also display slight negative returns, while January, March, and May exhibit moderate positive returns. December reflects the typical year-end rally. These patterns suggest that certain months provide better opportunities for gains, while others may present higher risks.

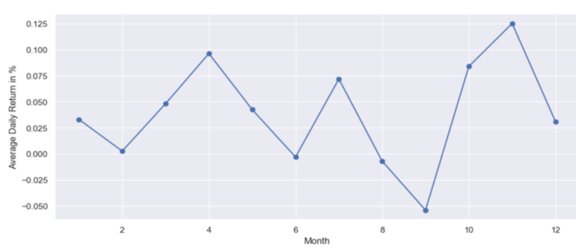


Fig. 8. Average daily return per month

#### 4) Average Return by Day of the Month

The average daily returns by day of the month reveal significant fluctuations in market performance. The 1st and 2nd days show the highest returns, suggesting a strong start to the month. Positive returns are also observed on days like the 13th, 16th, 25th, and 28th, indicating potential mid-month momentum. However, days such as the 9th, 19th, and 20th exhibit substantial negative returns, marking them as high-risk periods. The returns on the final days of the month tend to be slightly negative, suggesting weaker performance as the month closes. Overall, this pattern highlights potential opportunities early and mid-month, with caution needed around the 9th, 19th, and 20th.

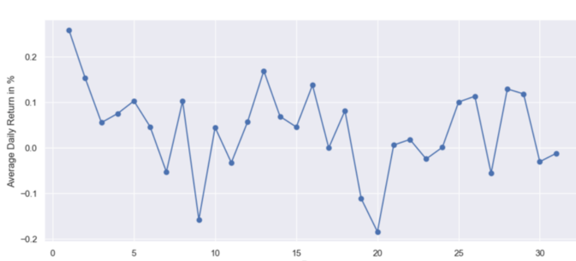


Fig. 9. Average daily return for each day of the month

#### 5) Returns and Volume Distribution Analysis

The 3D scatter plot shows the relationship between *20-day returns***,** *weekly relative volume***,** and *MACD***,** with color indicating the Target Dummy variable. Yellow points are clustered in the upper region, indicating positive returns, higher relative volume, and positive momentum. Blue points cluster in the lower region, reflecting negative returns and momentum.

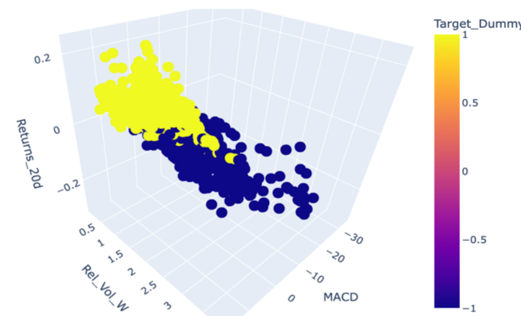


Fig. 10. Returns Distribution with Weekly Relative Volume and MACD

A clear downward trend between returns and MACD is visible, with negative MACD values linked to negative returns, while relative volume shows a weaker correlation to returns.

#### 6) Correlation Matrix

The correlation matrix reveals key relationships between financial indicators. Close prices are positively correlated with long-term returns and momentum, while negatively correlated with relative volumes. Short-term and long-term returns show moderate to strong correlations with each other, with momentum closely tied to medium-term returns. Volume shows a negative relationship with returns but a strong positive correlation with volatility, indicating that high volume often aligns with higher market volatility. Volatility measures are highly interrelated and negatively correlated with returns and momentum, suggesting that increased volatility often accompanies market downturns, whereas momentum signals positive price movements.

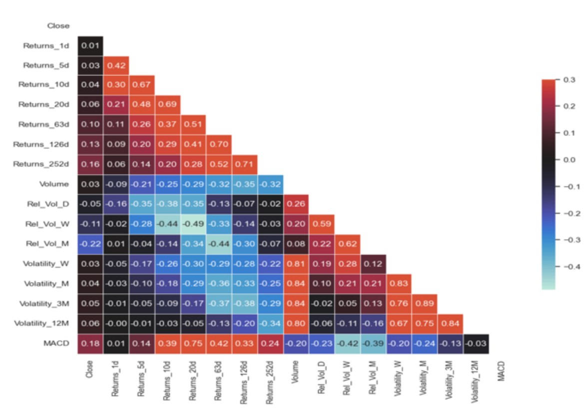


Fig. 11. Correlation Matrix

### E. Modeling

#### 1) Dimensionality Reduction using PCA

Principal Component Analysis (PCA) becomes especially important when dealing with high-dimensional datasets like this one with 50 variables. In this case, PCA has identified 10 components that explain 95% of the variance, which means that these 10 components capture almost all the essential information from the original variables. By reducing the number of features from 50 to 10, PCA simplifies the dataset without significant loss of information, making it easier to analyze and interpret.

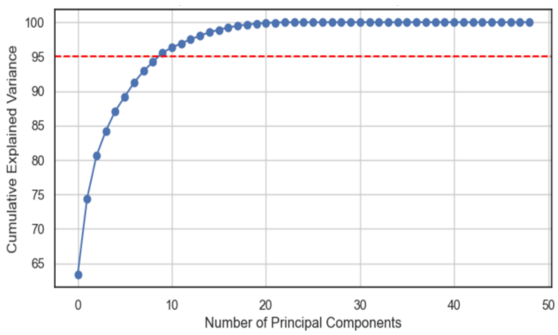


Fig. 11. Explained Variance Evolution with Number of Principal Components

This dimensionality reduction not only eliminates redundancy caused by correlated features but also helps improve computational efficiency and the performance of machine learning models. Additionally, by reducing the complexity of the dataset, PCA reduces the risk of overfitting, leading to more accurate and generalized predictions. In summary, PCA is crucial for handling large datasets, as it retains the core information while significantly simplifying the data.

#### 2) Data Partition

In this case, the 80/20 data partition is a beneficial choice as it provides a good balance between training and testing data. By using 80% of the data for training, the model has a substantial amount of information to learn from, while the remaining 20% allows for a robust evaluation of its generalization performance. This split is commonly used in machine learning because it ensures the model is not overfitted to the training data while still providing enough data for effective learning.

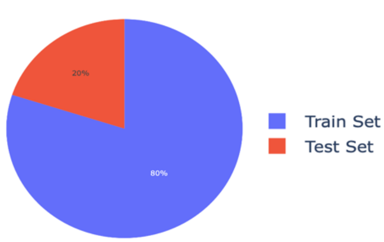


Fig. 12. Pie Chart Representation of our Data Partition

#### 3) Timeseries Split

Since the dataset involves time series data, where the chronological order of data points is crucial, it is important to implement a timeseries split instead of a traditional random split. A timeseries split respects the sequential nature of the data by dividing it into training and testing sets based on temporal order. This approach ensures that future data points are not used in the training phase, which mirrors real-world scenarios where predictions are made based on past data. Implementing this split can lead to more realistic and accurate performance evaluations, particularly when working with stock market data or any dataset where time dependencies are critical.

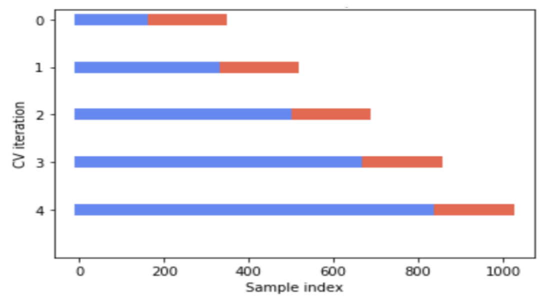


Fig. 13. Time Series Split for k=5

#### 4) Finding Optimal Hyperparameter:

The optimal hyperparameter finding process involves using *GridSearchCV* with *Time Series Split* to ensure the time series nature of the data is respected. For each model (*SVM, KNN, Naive Bayes, Random Forest, and MLPClassifier*), a parameter grid is defined to explore various hyperparameter combinations. *GridSearchCV* performs an exhaustive search over these combinations to identify the best hyperparameters based on cross-validated accuracy. After fitting the models using the training data, the best models are selected and evaluated on the test set, ensuring optimal performance and generalization.

- *SVM:*

Utilizes a regularization parameter *C* to balance model complexity and misclassification. Employs different *kernels*, for example linear or RBF, to manage data dimensionality and capture complex relationships.

- *KNN:*

Classification is based on proximity to neighboring data points, with sensitivity to the choice of *N Neighbors*.Weights can be adjusted to emphasize closer neighbors, enhancing decision-making accuracy.

- *Random Forest:*

Comprises multiple decision trees to improve prediction robustness and reduce overfitting through parameters like *N Estimators* and *Max Depth*. Employs minimum samples criteria to ensure generalization and effective tree growth.

- *MLP Classifier:*
Features multiple hidden layers with various activation functions to model non-linear relationships effectively. Uses optimization strategies, such as Adam and SGD, to minimize loss, with regularization techniques to mitigate overfitting.

## IV. RESULTS

The cumulative returns of our five different portfolio strategies over time, based on the following models: Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naive Bayes, Random Forest, and Deep Neural Network. Each strategy demonstrates different performance trends. The Deep Neural Network shows the highest cumulative return, consistently outperforming the others, especially post-2022. The SVM also performs well, maintaining strong returns and second to the DNN by 2024. Random Forest and KNN show similar trends, with steady growth but slightly less return compared to SVM and DNN. Naive Bayes, however, lags behind, with noticeable underperformance relative to the other methods, showing flatter returns after 2020. Overall, more complex models like DNN and SVM seem to yield better results over time compared to simpler methods like Naive Bayes.


Fig. 13. Comparative Returns of our Algorithms

The model has generally followed the overall upward trend of the asset price since 2020, successfully capturing many bullish periods, especially during the strong recovery after the COVID-19 dip in early 2020. The green sections align with periods of price increases, suggesting that the neural network was able to predict bullish momentum correctly in many cases. There are also red regions during downturns, particularly in 2022 when the market experienced volatility. Overall, the neural network seems to have adapted well to changes in market conditions, signaling bullish and bearish periods that correspond to significant price movements.

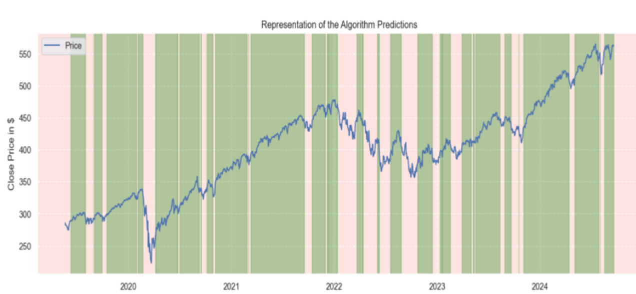


Fig. 14. Representation of Long and Short Period

The blue line represents the returns generated by the algorithmic strategy, while the brown line represents the buy-and-hold strategy. The algorithmic approach has clearly outperformed the buy-and-hold method, delivering significantly higher cumulative returns. From mid-2020 onwards, the algorithmic strategy shows a consistent upward trend, with returns nearing 200% by 2024, compared to around 50% for the buy-and-hold strategy.

The bottom graph shows the *drawdowns*, which represent the peak-to-trough decline in portfolio value, highlighting the risk and volatility of each strategy. The algorithmic strategy experiences less severe drawdowns compared to the buy-and-hold strategy. During market downturns, especially in 2020 during the COVID-19 market crash, the buy-and-hold strategy saw a maximum drawdown of over 35%, while the algorithmic strategy experienced less severe dips. Overall, the algorithmic strategy appears to manage risk more effectively, maintaining shallower and shorter drawdowns compared to the buy-and-hold approach.

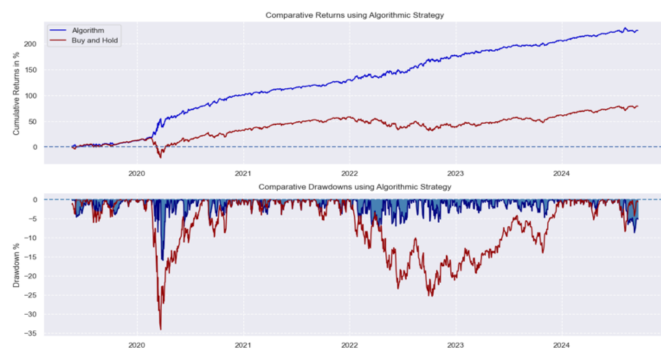


Fig. 15. Comparative returns and drawdown between the buy-and-hold and the algorithmic strategy

The annualized rolling Sharpe ratio of the DNN algorithmic portfolio provides valuable insight into the portfolio's risk-adjusted performance over time. With a *mean Sharpe ratio of 2.01*, the portfolio consistently delivers high returns relative to its risk. A Sharpe ratio above 1 is generally considered good, and a value exceeding 2 is exceptional, indicating that the DNN portfolio has been very efficient in generating returns for the level of risk taken. The *minimum value* of 1.12 shows that even during periods of underperformance, the portfolio still maintained a positive risk-adjusted return. The *maximum Sharpe ratio* of 3.09 highlights periods of outstanding performance, while the *25th and 75th percentiles* (1.73 and 2.28, respectively) indicate that most of the time, the portfolio operates in a highly favorable risk-return environment. The relatively low *standard deviation* (0.37) suggests that the Sharpe ratio is stable, with only moderate fluctuations. Overall, these statistics indicate that the DNN algorithmic portfolio has consistently provided strong, risk-efficient returns.

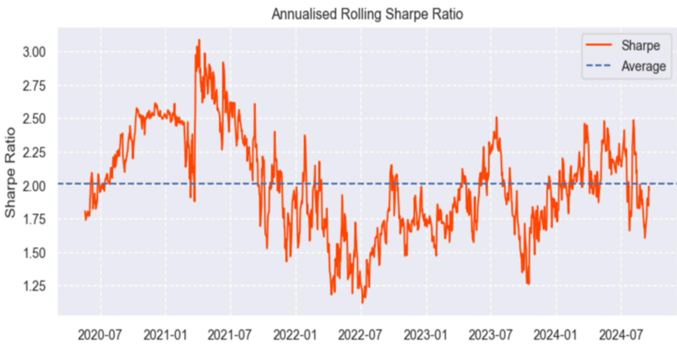


Fig. 16. Representation of the annualized rolling Sharpe ratio

The Monte Carlo simulation results provide a comprehensive understanding of the potential future performance of the strategy based on 3,000 random paths using Brownian motion. Over the next 252 trading days, the simulations show that the strategy's cumulative returns exhibit a range of possible outcomes, reflecting varying levels of confidence. At the *first standard deviation (68%),* the strategy is predicted to generate a performance between 30% and 50%, indicating that most likely the strategy will continue to yield positive returns. At the *second standard deviations (95%)*, the returns range from 10% to 75%, still covering a wide range but with a stronger assurance of positive outcomes. Even at the third standard deviations (99%), which represents extreme scenarios, the strategy is expected to yield returns between -5% and 86%, with only a slight chance of loss (up to -5%), showing robustness even in the most pessimistic cases. The *median performance* across all simulations is 40%, reinforcing that the strategy is generally expected to perform well under varying market conditions. These results suggest that the strategy is robust and has a high likelihood of delivering positive returns, with minimal risk of significant losses over the forecast period.

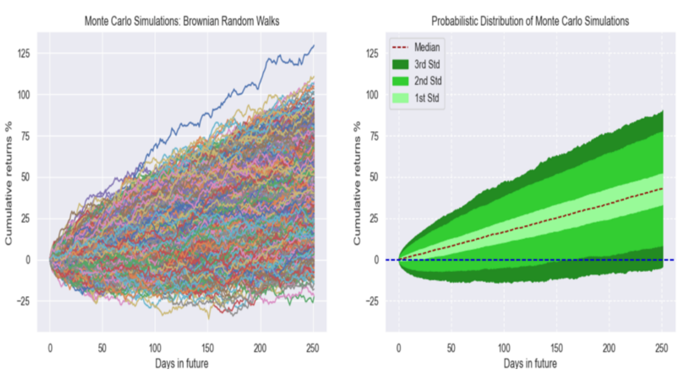


Fig. 17. Distribution of Brownian Random Walks

The analysis of the maximum drawdowns across 3,000 simulated paths over the next 252 days provides valuable insights into the potential downside risk of the strategy. The *mean maximum drawdown* is around -12.5%, which suggests that, on average, the strategy could experience a peak-to-trough decline of -12.5% during the forecast period. The standard deviation of drawdowns is 4.31, indicating moderate variability in the maximum drawdowns across different simulations. The minimum drawdown observed is -4.48%, implying that the most favorable scenario sees only a slight drop in value. On the other hand, the maximum drawdown is -37.98%, representing the worst-case scenario, where the strategy could potentially suffer a severe decline. The 25th percentile is -9.45%, and the 75th percentile is -14.72%, indicating that half of the simulated paths experience a maximum drawdown between these values. The median drawdown is -11.66%, suggesting that, in most cases, the strategy could face a moderate drawdown of around 12%.

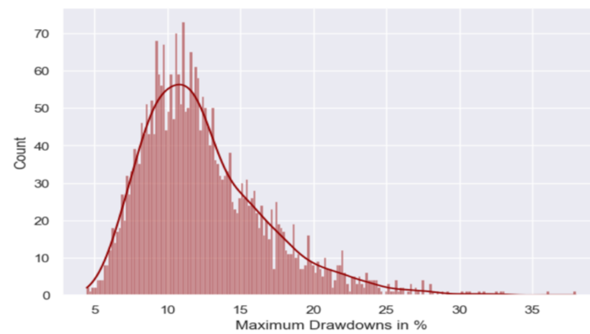


Fig. 18. Distribution of Maximum Drawdown of each Simulated Path

Overall, the results show that while the strategy generally maintains controlled drawdowns, there is a potential for more significant declines in extreme market conditions. However, the median and mean values imply that, under typical market conditions, drawdowns remain within a manageable range, reinforcing the robustness of the strategy.

## I. Conclusion

The DNN-based algorithmic trading strategy demonstrated superior performance compared to the buy-and-hold approach, both in terms of cumulative returns and risk management, as evidenced by higher Sharpe ratios and reduced drawdowns. The Monte Carlo simulations reinforced the robustness of the strategy, predicting favorable returns for the majority of future scenarios while keeping drawdowns within manageable limits. Despite the potential for extreme market conditions leading to significant drawdowns (up to 37.98% in the worst case), the average and median values indicate that the strategy is likely to perform well in typical market environments. Overall, the DNN algorithm provides a reliable method for maximizing returns while minimizing risk, making it a compelling choice for algorithmic trading.

## References


[1] Sharpe, W. F. (1975). "Likely Gains from Market Timing". *Financial Analysts Journal*, 31(2), 60–69.

[2] Clenow, A. (2013). "Following the Trend: Diversified Managed Futures Trading". *Wiley Trading*.

[3] Jegadeesh, N., & Titman, S. (1993). "Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency". *The Journal of Finance*, 48(1), 65–91.

[4] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques". *Expert Systems with Applications*, 42(1), 259–268.

[5] Appel, G. (1979). "The Moving Average Convergence Divergence (MACD)". *Systems and Forecasts*.

[6] Bollinger, J. (2001). "Bollinger on Bollinger Bands". *McGraw Hill Professional*.

[7] Fama, E. F., & Blume, M. E. (1966). "Filter Rules and Stock-Market Trading". *The Journal of Business*, 39(1), 226–241.

[8] Jolliffe, I. T. (2002). "Principal Component Analysis". *Springer Series in Statistics*.

[9] Metropolis, N., & Ulam, S. (1949). "The Monte Carlo Method". *Journal of the American Statistical Association*, 44(247), 335–341.

[10] Pardo, R. (2008). "The Evaluation and Optimization of Trading Strategies". *Wiley Trading*.