**AI-Based Support Decision Systems for Trade Operations in the Stock Market**

**Abstract**: The stock market is an important part of the world's financial system because it allows people to trade stocks, bonds, and other financial tools. This helps the economy grow by connecting investors with companies that need money. Even though new technologies like algorithmic and high-frequency trade have made things more efficient [6], the market's volatility and complexity are still big problems [43]. Machine learning and natural language processing tools like FinBERT [34], artificial intelligence (AI) has changed the way traders do business by letting traders look at huge amounts of data in real time [13]. This includes past trends, financial news, and market sentiment [18]. In this study, an AI-driven trading system is looked at that combines advanced sentiment analysis, real-time data collection, and automatic trade execution [45]. Backtesting shows that the system can do better than traditional benchmarks, giving good returns while managing risks in real time [7]. The results demonstrate that MLTrader outperforms the SPY benchmark in key risk-adjusted performance metrics, such as Sharpe Ratio (0.86 vs. 0.56) and Maximum Drawdown (0.66 vs. 0.44), highlighting its superior efficiency and risk management capabilities [3]. Even though there are problems with sentiment accuracy, adapting to a volatile market, and ethical concerns [35], future improvements—like using more data sources and better risk management—show how AI can change the way financial decisions are made, and dealing is done [26].

# **Introduction:**

## **Stock Market**

The stock market is counted among the great constituents of the world's financial system. It acts as a marketplace where people, corporations, and governments can deal in shares, bonds, and other financial instruments. It is basically a mechanism that connects investors with businesses or projects for the purpose of raising capital for the expansion and innovation of those businesses or projects. This exchange provides investors to improve their wealth while providing companies with funding for expansion, so it is a process that generates benefits to entire economy [40].

The value of the stock market goes above the financial gain. It supports economic growth. Companies that get capitals through the sale of shares or bonds use the funds to invest in innovative technologies, employ more people, or expand their services or operations. This improves productivity and promotes development. Investors use many strategies to predict the price fluctuations and identify investment opportunities in the stock market. These strategies include fundamental analysis, which weighs a company's financial stability. Technical analysis is the analyses of stock price charts and trends, and quantitative analysis applies data and mathematical models for predictions, the latter more widespread with progress in technology [39], [44].

The stock market has developed over time from simple local exchanges to an advanced global network. Billions of trades are made on well-known stock exchanges like NYSE, NASDAQ, and the London Stock Exchange each day [52]. These stock exchange platforms provide many ways that allow the traders and investors to take part in the world's financial markets. Technological advancement has drastically altered the way trading was made. The stock market today incorporates such innovations as electronic trading platforms, algorithmic trading methodologies, and high-frequency trading in a bid to get the trade quicker, efficient, and very accurate [2], [6].

The stock market has to face a lot of challenges all the time. It naturally follows an unpredictable path, depending on a lot of other factors such as economic indicators-including inflation, rate of interest, and unemployment rates-geopolitical events concerning political stability, conflicts, and trade policies [3], [4]. Investor sentiment shows the collective mood or emotion that can at any moment cause a stock market spike or crash [18].

This complexity has made it difficult for individuals to understand market behaviour. As a result, there is an increasing demand for the tools and methodologies that are able to process huge amounts of data, identifying trends, and generating information-driven predictions. This need has resulted in the introduction of Artificial Intelligence (AI) into the stock market [13], [28].

The main objective of stock market investing is to get capital returns and profits. The stock market and its trends are unstable in nature; researchers became attracted to predicting the next moves of this unstable market. The stock market operates according to its own principles, irrespective of the people who invest in or trade off. There are many factors that influence market fluctuations, making it difficult for companies and investors to make decisions. Market analysts and investors research market behaviour and plan their buy-or-sell strategies accordingly. Order books have always provided traders with the best possible ways in the traditional financial system. However, while it is functional and intuitive, things have changed significantly because of the quick pace at which technology is developing [16], [51]. As a result, the trading floor is now online, and the difference in the time between seconds and minutes can make an enormous loss or gain in the market. As the stock market produces a large amount of data every day, making it highly challenging for an individual to consider all the past and current information for predicting stock’s future direction [44], [58].

## **AI in stock market**

Artificial intelligence (AI) has become known as a most rapidly developing tool in the stock market, modifying the methodologies of investing and trading choices [13]. Artificial Intelligence refers to the use of machines and algorithms that learn from, adapt to, and make judgments from data. It involves AI handling huge volumes of data related to past prices, news of the economy and finances, trends in social media, among other parameters to draw inferences leading to perfect precision and speed for order execution in a stock market [26].

It brought complete changes into the trading processes where real-time analytics and the predicted outcomes seemed to have come up as unreachable fantasy earlier [16]. Automating in every trading procedure was hence introduced through systems that might monitor and adapt independently without interferences through people on it [6]. Not having this with manual analyses of AI basically grants one the chances for finding that could easily skip them [39].

The most important use of AI in the stock market is machine learning, where algorithms continually learn from an analysis of past data to become better predictors [12]. Machine learning algorithms might analyse historical fluctuations in prices with a view to predicting future patterns. These models are bound to get better over time as they get exposed to more and more data [13].

One of the broad areas of applications is natural language processing (NLP). With NLP, AI can read and understand human language; this, therefore, means analysing news stories, financial reports, and social media content for market sentiment [18]. This helps traders predict how future market behaviour will be to certain events by identifying the sentiment as either positive, negative, or neutral [54].

### **Impact of AI in stock market**

The combination of AI in the stock market has improved the trading methods and processes in decision-making [13].

Artificial intelligence has improved the productivity at the stock markets. With the ability to analyse a large amount of data in real time, AI makes traders to quickly respond to market fluctuations [26]. This reduces inefficiencies, improves liquidity, and ensures that tasks are performed with speed and accuracy. The operations of the market have grown more efficient and dependable the overall performance of market has become more efficient and dependable [6].

The main benefit is better predictability. Machine learning techniques changed the ball game of stock price fluctuation prediction [12]. These models explore historical data patterns and include real-time input to identify trends. In this way, it enables traders and investors to make accurate decisions with a high possibility of success in a competitive market [39].

AI automation has fully changed the way trading was performed. AI-based systems can themselves execute trades based on predefined rules and algorithms [6]. This eliminates human errors and brings consistency in strategies without emotional factors that generally lead to improper decisions [24]. Automated trading has made the stock market more accessible and less stressful for new traders [16].

Artificial intelligence has transformed risk management as well. By analysing historical data, market patterns, and investor sentiment, AI systems can identify possible risks and provide strategies for reducing risks [7]. AI can suggest stop-loss limitations, allowing traders to limit future losses. These strategies for risk assessment are growing as an asset for both individual and institutional investors [55].

Access to markets has been improved by artificial intelligence. AI-driven tools and platforms provide retail investors with personal advice and information previously reserved for financial institutions [13]. Access liberalization facilitates and helps individuals in making better financial decisions and participation in the market actively [16].

The ability of AI to perform sentiment analysis has changed the game in how traders perceive market sentiment. Through NLP and a number of methodologies, AI can analyse news headlines, social media involvement, and earnings reports for market sentiment [18]. This places traders in a very strong position to anticipate how markets will react to major news events, such as corporate announcements or geopolitical shifts [58].

There are some challenges that are facing AI in the stock market. Those are about ethics and regulation, especially in regard to high-frequency trading (HFT) [5]. While high-frequency trading can improve market efficiency, it may also result in uncertainty during volatility [2]. Challenges occur about the equity and accessibility of AI-driven systems, as well as the potential inequalities that come from the data they utilize [5]. Solving these concerns is essential for ensuring that AI improves the financial environment [20].

One of the features of AI is its capacity for adaptation and learning. AI systems improve their performance over time by processing new data and learning from previous errors [13]. This guarantees their continued relevance in a dynamic market situation and enables adaptation to rising problems and opportunities [26].

# **Objectives of Research**

* Develop an AI-driven trading system that integrates the FinBERT model for sentiment analysis of financial news headlines and executes trades based on the sentiment classification (positive, negative, or neutral).
* Implement a trading strategy using the Alpaca API for real-time data retrieval and trade execution, with sentiment analysis guiding buy/sell decisions.
* Evaluate the trading system's performance using backtesting on historical data (e.g., SPY ETF) and metrics such as total return, Sharpe ratio, and maximum drawdown.
* Design a dynamic position-sizing mechanism that adjusts trade quantities based on available cash and predefined risk parameters (e.g., cash-at-risk).
* Demonstrate the system's ability to adapt to market conditions by continuously monitoring sentiment trends and executing trades with high confidence levels (probability > 0.999).

# **Literature Review**

Financial market research has adopted sentiment analysis and text mining as techniques for making trading decisions [37]. Patric et al. researched sentiment mining from German stock market news with the SentiWS tool [37]. Their methodology linked textual sentiment data with stock price patterns to offer practical daily recommendations, providing investors with techniques to manage risks and improve decision-making.

Shynkevich et al. examined the utilization of multiple kernel learning (MKL) in processing financial news [39]. The integration improved the accuracy of trend predictions, achieving 79.59% with the usage of polynomial kernels. Their findings exposed the disadvantages in traditional algorithms such as SVM and k-NN, which were shown to reduce predictive efficiency [39].

Ho’ang and Phayung developed a hybrid prediction approach that integrated Vietnam stock index data with sentiment derived from news articles [37]. This study was limited to closing price data, indicating potential for improvement by incorporating more varied data sources.

Jageshwer and Shagufta focused on the impact of financial news on stock prices and daily market fluctuations [38]. Their research combined technical analysis with a rule-based classification method, employing financial news and average monthly stock prices to improve prediction results.

Ruchi and Gandhi stated the importance of natural language processing (NLP) in analyzing stock trends [33]. The study offered market sentiment insights but recognized the need for additional input variables and advanced algorithms for better accuracy in predictions.

Abdullah et al. examined the Bangladesh stock market, utilizing text mining and natural language processing techniques to derive financial insights [45]. By incorporating essential measures like earnings per share (EPS) and price-to-earnings (P/E) ratios alongside historical stock data, their algorithm generated definitive buy or sell signals. Apache OpenNLP simplified text processing, allowing for a comprehensive analysis of market-related textual data [44].

These studies indicate advancements, but forecasting accuracy for stock market patterns frequently falls short of 80%, with sentiment-driven models typically achieving lower standards [39]. Neural networks and machine learning methodologies have been beneficial but have not consistently reached high levels of accuracy [45].

The current study hence intends to fill these lacunae by developing models that integrate sentiment analysis with historical financial data. Such methods employ state-of-the-art NLP and machine learning algorithms to further enhance forecast accuracy, thereby enabling investors to make more informed decisions [47]. The approach aims to create a profitable financial market environment by mitigating risks and optimizing returns [46].

# **Approach and Methodology**

## **Methodology**

*Figure-2.2.1: Methodology of AI based decision making system.*

This section highlights the steps in the methodology of the AI-driven trading decision system. As discussed earlier, the entire technique is fragmented into various steps so that clarity, efficiency, and accuracy are provided all at once. The proposed system embodies data gathering, sentiment analysis, risk assessment, decision-making, trade execution, and performance monitoring within a single framework of an adaptive trading system [48].

* **Step 1: Retrieving the Market Data and News**

The first step in the AI trading system is collecting all the necessary information, which emanates from many sources. These include:

**Market Data Acquisition**: The system makes a connection with the Alpaca API to allow it to get both real-time and historical financial data [46]. These include important information such as stock prices, trade volumes, and historical patterns that provide insights into market behaviour, enabling the system to predict future trends [44].

**News Data Retrieval**: This system collects news headlines applicable to the selected trade symbol; for example, SPY represents an ETF for the S&P 500 [16]. News headlines are often employed as indicators of market sentiment, encompassing news about economic developments, business performance, and geopolitical factors [30]. These headlines play a crucial role in the sentiment analysis model, integrating both quantitative and qualitative elements in trading decisions [23].

* **Step 2: Perform Sentiment Analysis**

System after collection of news information proceeds to process all collected data to obtain meaningful sentiment insight which includes the following:

**Text Tokenization**: The system executes tokenization of the news headlines using the AutoTokenizer library by Transformers [50]. Tokenization is the process of breaking down text into smaller units for processing before any machine learning model takes effect [19].

Sentiment Classification: FinBERT is pre-trained with financial text, and it classifies every tokenized headline to positive, negative, or neutral [33]. It is how market sentiment would be viewed from news events [18].

Model Fine-Tuning: Though a pre-trained model, finetuning can be done for each specific trading scenario or dataset, ensuring the best performance in the selected context of finance [36]. It would use qualitative insights in its decisions by how it structures sentiment outputs from unstructured textual data [22].

* **Step 3: Calculate Sentiment Scores**

Following classification, the system will attach a confidence score for sentiment analysis between 0 and 100 to show how confident the model is with respect to the classification of that particular text:

High Confidence Scores: Under such conditions, confidence should be above a certain percentage; for instance, 99.9%. Only then can the sentiments be trusted [33].

Low Confidence Scores: Less than the required threshold is discarded because the trust is not there in the decision process to action [12]. Sentiment scores take the qualitative data of sentiment and make it quantitative while linking it with actionable trading signals, adding this level of reliability to the system [28].

* **Step 4: Assess Risk Parameters**

Risk management is essential to the system's methodology. Before performing any trades, the system checks the level of risk with the following:

Cash allotment: The system specifies how much of the cash available would be risked on a trade. For instance, this may go up 50% of the total cash to let the potential gains and losses remain comparable [16].

Position sizing: From cash allocation and the existing stock price, this set of calculations provide the number of shares to be traded. This ensures the available resources and the size of the position on the trade are within the risk set tolerance of the system [21].

Operating within a fixed parameter of risk, the system lessens the number of huge losses.

* **Step 5: Making Trade Decisions**

The trading actions this system has to effect are determined from the knowledge gained from sentiment analysis and the assessment of risk. The decision-making process includes:

Buy Decisions: If the confidence score is high. If the emotion is positive for sentiment, then automatically this system issues a buy order, expecting a rice in the price [13].

Sell Decisions: If the sentiment is negative, but the confidence score comes to be high, then this system will place a sell order expecting a decline in the price[24].

Hold Decisions: In such a case, when there is either a neutral sentiment or low-confidence scores, no action is taken with the view of helping the system avoid general risks that may not be necessary [45].

* **Step 6: Trade Execution**

In case of a trading decision, the system will execute the trade via Alpaca API. The execution also entails the bracket orders of risk mitigation:

Take-Profit Orders: It is a pre-defined price level at which it automatically sells to lock cash in profit [46].

Stop-Loss Orders: A stoploss is a pre-defined price level where the system exits the trade to cut losses beyond a certain level [7].

Bracket orders act as safety nets to protect trades from extreme market fluctuations [6].

* **Monitor Performance and Adjust Parameters:**

This performance-monitoring activity is very crucial in ensuring that the trading system is still efficient. The system will then keep monitoring even other key performance indicators such as:

The return on investment (ROI): It measures how profitable the trading strategy is [9].

Drawdowns: the maximum observed loss from peak to trough [11].

Sharpe Ratio: Risk-adjusted returns measure [32]

The system dynamically adjusts parameters such as sentiment confidence thresholds, risk allocation percentages, and stop-loss or take-profit levels to remain adaptive to changing market conditions. These refinements enable the system to effectively respond to new challenges and capitalize on emerging opportunities [14].

## **Tools**

* **Libraries and Frameworks:**

**Transformers by Hugging Face:** It is currently the most popular library for every aspect of NLP. It provides state-of-the-art pre-trained models but also utilities to preprocess the text data [33]. The AI trading system will apply the Transformers library for loading and manipulating the FinBERT model. This model specializes in a financial sentiment for trading decision making. Using the library, news headlines are tokenized into a model-understandable format, processed into meaningful information, and classified into positive, negative, or neutral feelings This classification directly affects the trading decisions made by the strategy, forming a primary component of the architecture [33].

**PyTorch**: PyTorch is generally regarded as the open-source library that can be used for building deep learning models. Significantly, this system thereby avails adept computation and managing of the FinBERT model [46]. The tokenized input will then be fed through the FinBERT model for scoring the probability of sentiment categories using PyTorch [33]. Besides, it will capably put the system in a position of processing massive volumes of financial news in real time with the trading environment demands due to the fact that it can be accelerated by GPU [31]. This makes this system an important capability into keeping the trading strategy flexible to dynamic changing market circumstances.

**Lumibot:** Lumibot is an algorithmic trading library designed with Python. It provides a simple approach to developing, testing, and deploying trading strategies [29]. Quite simply informed trade algorithm acts as intermediaries between the trading algorithms and the Alpaca broker, downloading data, executing trades, or monitoring performance [ 29]. Besides, it allows for backtesting through which the strategy would be tested with historical data in a strict manner. Because of that, the system would be secured for reliability and optimization before being applied in real-time markets [55]. Such capabilities of integration and structured methodology aspect make Lumibot an essential tool in setting up this AI trading system.

**Datetime:** This is a standard library initialized in Python for date and time standardization and manipulation. Its utility in this system sets the trading window which entails, for instance, the last three days for news data to be analyzed for sentiment. It makes the iteration process of trading synchronous with real-time events, so that trades can happen based on the current state of the market. This level of precision in scheduling day and time helps keep the analysis and decisions of the system in point and relevant [9].

* **APIs:**

**The Alpaca Trading API:** It presents a strong basis of systems which has all functionalities that require market data, financial news retrieval, and trades execution under the condition of paper or live trading [53]. Under this, the entire test can be performed before going live as with the majorly used features in this system that is bracket orders, which automate the risk management measures by setting predefined take-profit and stop-loss levels. That would ensure dynamic trades to manage reduction of risks and optimize the returns. Further, seamless integration of Alpaca API with Lumibot would serve to further streamline the entire trading operations [29].

**Yahoo Finance (by Lumibot):** Yahoo Finance can be used for historical market data sources which can be accessed through the Lumibot framework for back testing purposes. Historical data is critical for simulating the trading strategy in all kinds of markets such as bullish, bearish to very volatile markets. After testing the strategy's performance in these scenarios, the approach has to bring further refinements to adapt to the real-world conditions. This data-driven trend will guarantee the trading scenario to be very strong and flexible [29].

* **Main Tools and Functionalities**

**FinBERT Model:** The FinBERT model is a slice of BERT, customized for driving its performance to always perfection in sentiment analysis for the 'domain' of finances. This forms the basis for being able to headline a system that can give understanding of news headlines [33]. This model derives tradable insights from the understanding of sentiment in headlines. A simple example would be to buy on positive sentiment and sell on negative tweets [18]. Since it is trained on financial texts, the model stands invaluable to this trading system as it has ensured high accuracy in understanding complex financial language and jargons that would critically pare down this trading system's profitability [30].

**Torch CUDA:** Torch CUDA fully exploits GPU for the enhancement of computational efficiency in PyTorch framework [31]. In the current setup, CUDA is used in the application for speeding up the mechanism of sentiment analysis done on huge amounts of news data processing in real-time. It becomes extremely important in a trading environment where the delay in processing will lose precious time and lead to a loss in profits [21]. Then, if all the conditions are successful, CUDA primarily speeds up and adds value to the overall performance of this trading system with computational efficiency.

**Bracket Order (using Alpaca API):** Bracket orders are an advanced feature provided by the Alpaca API to automate risk management [22]. They let the system define a take-profit price and a stop-loss price for each buy/sell transaction. If the price reaches a certain level, the transaction is closed to harvest profits; on the other hand, if the price reaches a predetermined loss threshold, it is closed to assure losses. This has been installed in the trading strategy for the trade to manage itself without any person to intervene in the process or the overall control of risk [7].

* **Development Environment**

**Python:** This is the programming language that is used in developing the trading decision system based on the simplicity, versatility, and rich library ecosystem in Python [46]. Python's capability to collaborate with machine learning libraries, APIs, and data-processing resources make for an ideal choice if one wants to develop a highly complicated trading system [13]. Besides this, Python has clear syntax that makes teamwork and further code maintenance easy [25].

**IDE/Code Editor:** An integrated development environment (IDE), or a code editor that provides the essential system. It is the case with Visual Studio Code [49]. Its functionalities of syntax highlighting and code linting assure coding more comfortable, else other features of possibly debugging with point-by-point breakdown [17].

**Backtesting Framework (Lumibot):** The backtesting framework offered by Lumibot serves to measure the efficacy of trading model [21]. This simulation execution of trades in historical market data for identification of the issues helps to figure out and refine some of those things before they go live. Durning backtesting, metrics like profit, drawdown, and Sharpe ratio facilitates identifying suitable modifications and improvements to the trading strategy [7].

# **Analysis**

## **Data Flow Analysis**

The below figure-5.1.1, represents the workflow of an AI-Based Support Decision Systems for Trade Operations in the Stock Market, illustrating how data moves between different components, from gathering market data to executing trades and analyzing performance. The Market Data Source provides financial data through API Integration, which is then processed in the MLTrader Class Initialization to develop a Trading Strategy [13]. Position Sizing determines the appropriate investment for each trade, while Sentiment Analysis Integration enhances decision-making by incorporating external sentiment data [14, 18]. The Trading Iteration Logic executes trades and logs the results. Finally, Backtesting evaluates past performance and generates Backtesting Results for the User to analyze, ensuring continuous improvement of the trading strategy [55].

A diagram of a flowchart

Description automatically generated

Figure-5.1.1: Data Flow Diagram of AI-Based Support Decision Systems for Trade Operations in the Stock Market

**Market Data Source:**

As shown in the figure-5.1.1, the Market Data Source is an external entity that forms the basis of the AI-based trading system, providing the necessary real-time market data in terms of stock prices, trading volumes, and financial news headlines [26]. This data is primarily useful for the system to make informed and timely trading decisions, as it guarantees the availability of accurate and up-to-date information, which is then to be analyzed for market trends, identifying trading opportunities, and executing trades effectively [44]. Without robustly accurate market data, the system would not have any input to generate trade signals meaningfully [51]. Then comes the stage of API Integration Processing, where data from the Market Source flows smoothly. In here, data is fetched from the data source and prepared for subsequent processing [42]. This guarantees the data is sufficiently prepared for the trading strategy and sentiment analysis components so as to guarantee the system maintains efficacy and responsiveness with respect to the market changes [13]. The Market Data Source can include several major financial data providers, mainly Bloomberg, Reuters, and specialized APIs like Alpha which provide not only real-time data but also historical ones for backtesting and performance evaluation [26, 51]. The quality and reliability of the Market Data Source have a direct bearing on how accurate the trading system's decision-making will be, making the selection of a strong and reputable data provider an important step in constructing a successful AI-based trading system [44].

**Establishing Connection to the Trading Platform (API Integration):**

In Figure-5.1.1, the process of API Integration acts as a bridge linking the Market Data Source to the Trading System, where market data flows from the Market Data Source toward API Integration [42]. At this point, data is being fetched and prepared for further analysis, after which it is rendered to the Trading Strategy and Sentiment Analysis component [44]. This seamless flow of data serves in the generation of trading signals and execution of trades via the Alpaca API, providing efficient functioning and fast response to changes in the market [59]. This configuration allows the system to engage the Alpaca API for live trading and paper trading, hence securing accessibility to market data, trade execution, and portfolio management [6]. Provided that it uses the Alpaca API, the users can build, back-test, and deploy their trading strategies with full confidence, ensuring sound performance in real-life trading situations [55].

A screen shot of a computer code

Description automatically generated

*Figure-5.1.2: Code snippet of Integration of Alpaca API*

In Figure-5.1.2, the API\_KEY is a secret key tied to the user's account, authorizing requests to the Alpaca platform. This key is user-specific and must be kept private to ensure security, while the API\_SECRET acts as a password, verifying the user's identity for each request. Together, the API\_KEY and API\_SECRET ensure that only authorized individuals can access the trading account [51]. The BASE\_URL is configured to point to the Alpaca paper trading API, a virtual environment that allows traders to test strategies without risking actual capital. Paper trading is an invaluable tool for refining skills and testing strategies in a low-cost or no-risk setting [6]. The credentials (API\_KEY, API\_SECRET, and PAPER) are stored in a dictionary (ALPACA\_CREDS), which centralizes the configuration and makes it easily accessible throughout the trading system [51]. With this setup, the system can interact with the Alpaca API for both live trading and paper trading, enabling secure access to market data, trade execution, and portfolio management [59]. This integration allows users to build, test, and deploy trading strategies with confidence, ensuring robust performance in real-world trading scenarios [55].

**MLTrader Initialization and strategy:**

The trading system can freely fetch market data and execute trades using the Alpaca API. As shown in Figure 5.1.1, the symbol parameter specifies the asset to be traded, for example, "SPY." This sets the fraction of available cash to be put at risk for any trade [13]. This initialization process ensures that the trading strategy is ready to process market data, develop trading signals, and execute trades, denoted by the data flow from API Integration to Trading Strategy in Figure 5.1.1 [44]. The initialization of the MLTrader class builds upon API integration by using the configured credentials to establish a connection to the Alpaca API, allowing the trading system to readily access real-time market data to execute trades efficiently [42].

*A computer code with text

Description automatically generated*

*Figure-5.1.3: code snippet of MLTrader Initialization*

The MLTrader class in Figure 5.1.3 is initialized and inherits from the Strategy base class. On calling the initialize method, the baseline parameters are set: self.symbol specifies the trading symbol (example: "SPY"), self.sleeptime sets the trading interval to '24H', self.last\_trade tracks the type of the last trade executed (initialized to None), and self.cash\_at-risk specifies the fraction of available cash to be risked per trade [13]. The self.api attribute creates a connection to the Alpaca API using the credentials established in the previous step (5.1.2) [51]. Following initialization, the trading system is ready to fetch market data and execute trades against the Alpaca API [59].

**Position Sizing**

As shown in Figure-5.1.1, the position sizing function of the MLTrader class is one of the key functions used to determine the number of shares to trade based on available cash, the last price of the stock, and a set risk parameter [44]. The cash and stock price parameters connect from Step 5.1.2, API Integration, and Step 5.1.3, MLTrader Initialization, where the API credentials and core trading parameters were set up [42]. In Figure-5.1.4, the position\_sizing method computes the optimal trade size by dividing the available cash for risk (cash \* self.cash\_at-risk) with the stock's last price [7]. Such an approach will comply with the risk management framework of the strategy for every trade while providing flexibility according to the changes in market parameters and account balance [32]. The data flow proceeds from the Trading Strategy process to the Trade Execution process, as shown in Figure-5.1.1, using the trade size derived from the calculated trade size to execute trades snugly [51]. That makes for a seamless resolution to ensure that trading strategies run accurately and reduce risk exposure [55].

A computer code with text

Description automatically generated with medium confidence

*Figure-5.1.4: Code snippet of Position Sizing*

As shown in Figure-5.1.4, the position sizing method starts by accessing the trading account's available cash via self.get\_cash() and the current market price of the stock with self.get\_last\_price(self.symbol). These two values assure that the calculations use correct or actual data [7]. Then the trade size is calculated via the expression quantity = round(cash \* self.cash\_at-risk / last\_price, 0), which divides the available cash for risk over the stock's last price and rounds it to the nearest integer [32]. For example, with $10,000 available cash, a stock price of 100, and a cash-at-risk of 0.5, the method calculates an optimal trade size of 50.

**Sentiment Analysis Integration:**

As illustrated in Figure-5.1.1 (DFD), the Sentiment Analysis process is a critical component that analyzes financial news headlines to determine market sentiment, which is then used to inform trading decisions [18]. This step connects directly to the Sentiment Data and Position Sizing processes, where sentiment data influences the calculation of trade sizes and risk parameters [33]. In Figure-5.1.5, the estimate\_sentiment method utilizes the FinBERT model to classify the sentiment of financial news headlines as positive, negative, or neutral, along with a confidence probability [33]. The sentiment data flows from the Sentiment Analysis process to the Trading Strategy process, where it is combined with position sizing data to generate trading signals [14]. This integration ensures that the trading strategy is responsive to external market factors, enhancing decision-making accuracy and adaptability [18].

The estimate\_sentiment method, as shown in Figure-5.1.5, begins by checking if the input contains any news headlines. If no headlines are provided, it returns a neutral sentiment with zero confidence [33]. For valid input, the headlines are tokenized using the FinBERT tokenizer, converting the text into tensors suitable for model inference [33]. The tokenized data is then passed through the FinBERT model to generate raw sentiment logits, which are normalized into probabilities using the SoftMax function [33]. The method identifies the sentiment category with the highest probability and extracts the corresponding confidence score and sentiment label [33]. These results are returned to the Trading Strategy process, where they are combined with position sizing data to generate trading signals [14]. This seamless integration ensures that the trading strategy is responsive to external market factors, enhancing decision-making accuracy and adaptability, as depicted in the data flow of Figure-5.1.1 [18].

**Sentiment Data**

As illustrated in Figure-5.1.1, the get\_sentiment method is a critical component that integrates sentiment analysis into the trading strategy, enabling it to make informed decisions based on external market factors [14]. This step connects directly to the previous steps, where the API credentials, core parameters, and trade size calculations were configured [51]. In Figure-5.1.5, the get\_sentiment method fetches recent news headlines related to the stock symbol (self.symbol) using the Alpaca API and performs sentiment analysis to determine whether the sentiment is positive, negative, or neutral, along with a confidence score [33]. This sentiment data is then used by the trading strategy to make informed decisions, as shown in the data flow from Sentiment Analysis to Trading Strategy in Figure-5.1.1 [14]. By incorporating external market sentiment, the strategy ensures that trades are aligned with prevailing market conditions, enhancing decision-making accuracy and adaptability [18].

A computer code with text

Description automatically generated

*Figure-5.1.5: Sentiment Analysis Integration*

The get\_sentiment method, as shown in Figure-5.1.5, begins by setting a date range using self.get\_dates() to fetch recent news headlines [33]. It then uses self.api.get\_news() to retrieve news articles related to the stock symbol within the specified date range [51]. The headlines are extracted using a list comprehension and passed to the estimate\_sentiment function, which analyzes the headlines and returns two values: probability (a confidence score) and sentiment (positive, negative, or neutral) [33]. For example, a positive sentiment with a high probability (e.g., 0.98) might trigger a buy signal, while a negative sentiment could prompt a sell signal [14]. The method returns these values, allowing the trading strategy to incorporate sentiment-driven insights into its decision-making process [18]. This integration ensures that the strategy remains responsive to external market factors, enhancing its accuracy and adaptability, as depicted in the data flow of Figure-5.1.1 [33]**.**

**Trading Iteration Logic:**

As illustrated in Figure-5.1.1 (DFD), the on\_trading\_iteration method is the core of the trading strategy, where market conditions, sentiment, and risk parameters are evaluated to execute trades [13]. This step connects directly to the previous steps, where API credentials, core parameters, trade sizes, and sentiment data were configured [51]. In Figure-5.1.6 and Figure-5.1.7, the on\_trading\_iteration method uses the outputs from position\_sizing (cash, last price, and quantity) and get\_sentiment (probability and sentiment) to make trading decisions [32]. The data flows from the Trading Strategy process to the Trade Execution process, as shown in Figure-5.1.1, where buy or sell orders are executed based on sentiment and risk management rules [14]. The results of these trades are logged in Trade Logs, and performance metrics are sent to the User and Backtesting Results, ensuring a closed-loop system that adapts to market changes and maintains disciplined trading [55].

A screen shot of a computer code

Description automatically generated

*Figure-5.1.6: code snippet of Trading Iteration Logic: Positive Sentiment*

The on\_trading\_iteration method, as shown in Figure-5.1.6 and Figure-5.1.7, begins by retrieving the available cash, last price, and trade quantity using self.position\_sizing() and fetching the sentiment and confidence score using self.get\_sentiment() [18]. If the sentiment is positive and the confidence score exceeds 0.999, the method closes any open sell positions and creates a buy order with a take-profit price at 20% above the current price and a stop-loss price at 5% below [7]. Similarly, if the sentiment is negative with a confidence score above 0.999, it closes any open buy positions and creates a sell order with a take-profit price at 20% below the current price and a stop-loss price at 5% above [7]. These orders are submitted using self.submit\_order(order), and the self.last\_trade attribute is updated to reflect the latest trade action [51]. This method ensures disciplined and risk-aware trading by integrating sentiment analysis, positionsizing, and risk management, as depicted in the data flow of Figure-5.1.1 [32].

**A screen shot of a computer code

Description automatically generated**

*Figure-5.1.7: code snippet of Trading Iteration Logic: Negative Sentiment*

**Trading Logic**:

As illustrated in Figure-5.1.1, the Trading Logic is the central component that integrates Backtesting and Trading Iteration Logic to ensure a cohesive and adaptive trading strategy [13]. The Backtesting process evaluates the strategy using historical market data, measuring key performance metrics such as Total Return, Maximum Drawdown, and Sharpe Ratio to assess its effectiveness and adaptability [55]. The data flows from Backtesting to Trading Logic, where these metrics are analyzed to optimize parameters such as risk settings (cash\_at\_risk) and sentiment thresholds (probability > 0.999) [7]. These optimized parameters are then passed to the Trading Iteration Logic, which executes trades in real-time based on sentiment analysis, position sizing, and risk parameters [14]. For example, if the sentiment is positive with a confidence score exceeding 0.999, a buy order is created with a take-profit price at 20% above the current price and a stop-loss price at 5% below [7]. Similarly, for negative sentiment, a sell order is created with a take-profit price at 20% below the current price and a stop-loss price at 5% above [7]. The results of these trades are logged in Trade Logs, and performance metrics are sent to the User and Backtesting Results, ensuring a closed-loop system that adapts to market changes and maintains disciplined trading [55]. This seamless integration of Backtesting, Trading Logic, and Trading Iteration Logic ensures that the strategy operates efficiently and responds effectively to real-time market conditions [13].

**Backtesting:**

As illustrated in Figure-5.1.1, the Backtesting process is a critical step that evaluates the trading strategy's performance using historical market data [55]. This step connects directly to the previous steps, where the strategy's core components were configured and tested [51]. In Figure-5.1.8, the backtesting setup defines a framework to evaluate the strategy's performance over a specified historical period, from January 1, 2020, to December 31, 2024 [55]. The broker is initialized using the Alpaca API credentials stored in ALPACA\_CREDS, and an instance of the MLTrader strategy is created with the name and configured parameters (symbol: "SPY", cash\_at\_risk: 0.5) [51]. The backtesting process is executed using strategy.backtest(...), leveraging the historical data provided by the YahooDataBacktesting framework [55]. This process evaluates the strategy's performance across varying market conditions, measuring key metrics such as Total Return, Maximum Drawdown, and Sharpe Ratio, as shown in the data flow from Backtesting to Backtesting Results in Figure-5.1.1 [7].

A screen shot of a computer code

Description automatically generated

*Figure-5.1.8: Code snippet of Backtesting the Trading Strategy*

The backtesting process begins by setting the start\_date and end\_date to define the evaluation period [55]. The broker is initialized using Alpaca API credentials, enabling access to historical market data [51]. An instance of the MLTrader strategy is created with specific parameters, including the stock symbol (SPY) and the percentage of available cash to risk per trade (50%) [13]. The strategy.backtest(...) function is then executed, leveraging the YahooDataBacktesting framework to evaluate the strategy's performance over the defined period [55]. Key performance metrics, such as Total Return, Maximum Drawdown, and Sharpe Ratio, are generated to assess the strategy's effectiveness and adaptability to historical market conditions [7]. These metrics are logged in Backtesting Results and sent to the User, as shown in the data flow of Figure-5.1.1, ensuring a comprehensive evaluation of the strategy's performance and providing insights for optimization [55].

**BackTesting Results:**

As illustrated in Figure-5.1.1 (DFD), the Backtesting Results are a critical output of the trading system, providing a comprehensive evaluation of the strategy's performance using historical market data [55]. This step connects directly to the Backtesting process, where the strategy is tested over a specified historical period (e.g., January 1, 2020, to December 31, 2024), and to the Trading Iteration Logic, where real-time trading decisions are made [13]. The data flows from the Backtesting process to the Backtesting Results, where key performance metrics such as Total Return, Maximum Drawdown, and Sharpe Ratio are calculated and stored [7]. These metrics provide insights into the strategy's effectiveness, risk-adjusted returns, and adaptability to historical market conditions [55]. The Backtesting Results are then used to optimize the strategy's parameters, such as risk settings and sentiment thresholds, which are passed to the Trading Iteration Logic for real-time trading [14]. The Trading Iteration Logic executes trades based on sentiment analysis, position sizing, and risk parameters, ensuring that the strategy adapts to changing market conditions [18]. The performance metrics from the Backtesting Results and the real-time trading results from the Trading Iteration Logics are sent to the User, providing detailed insights into the strategy's performance [55]. This closed-loop system ensures that the strategy is continuously refined and adapted to achieve optimal performance, as depicted in the data flow of Figure-5.1.1 [13].

**User:**

As illustrated in Figure-5.1.1, the User is the external entity that interacts with the trading system, receiving performance metrics and insights generated by the Trading Iteration Logic and API Integration processes [13]. The API Integration process fetches real-time market data and executes trades, while the Trading Iteration Logic generates trading signals based on sentiment analysis, position sizing, and risk management [14]. The data flows from the Trading Iteration Logic to the User, providing detailed performance metrics such as Total Return, Maximum Drawdown, and Sharpe Ratio [55]. Additionally, the API Integration process ensures that the user has access to real-time market data and trade execution capabilities, enabling them to monitor and manage their trading strategy effectively [51]. This seamless flow of data ensures that the user remains informed about the strategy's performance and can make adjustments as needed to optimize results [13].

## **Results:**

Following successful development and testing of AI-based strategy trading decision system, it was back-tested between January 2020 and December 2024. In this test, account was taken in examining the tracking system's decision-making quality, risk management efficiency, and the possibility of profit in general [13]. The AI system is capable of providing trading recommendations as programmed in it. It works on the basis of sentiment analysis as well as financial data. This analysis is done with the help of advanced machine learning models and API integration [18].

Given the backtest, two distinct quantitative performance sheet outputs were generated: one being the Tearsheet, and two are an interactive plot file, providing significant analysis of how well the system performed. This output is used to compare the trading strategies set up and above, and to inform on how well it fared against SPY, an ETF that tracks the S&P 500 [2].

**Interactive Graph File:**

The below figure-1, demonstrates the cumulative returns of the AI strategy when compared to SPY, thus goes to show that the system has made more profits than SPY over the evaluation period [21]. All the buys and sells are shown as markers so that users can actually work out when the trades were carried out. The cash flow segment also provides showing a distribution of funds throughout the trading period. All these graphs are generated automatically, as Lumibot incorporates with Python libraries [29]. These visual outputs are generated immediately within the framework during a backtest, eliminating the need for specific coding [11].

A screen shot of a graph

Description automatically generated

*Figure-3.2.1: MLTrader Strategy Compared with SPY*

The figure-1 provides a comprehensive visualization of the performance of the AI-based trading decision system (MLTrader Strategy) compared to the SPY ETF, a benchmark representing the S&P 500, over the period from 2020 to 2025. The following are the key components of the Graph:

**Lines Representing Performance:** The blue Line (MLTrader Strategy) represents the cumulative value of the MLTrader Strategy over time, including gains and losses from executed trades. While, Orange Line (SPY) line shows the performance of SPY, the benchmark, over the same period. It provides a baseline for comparison, demonstrating how the market performed overall [45]. The green line shows the cash position of the trading system over time. It indicates how much money was retained as cash, rather than being invested in trades [5].

**Buy and Sell Markers:** Green Upward Triangles represent points where the MLTrader system executed buy trades, signalling entry into a position. Red Downward Triangles indicate sell trades, where the system exited a position to either lock in profits or minimize losses [10].

**Dual Axes:** The left axis represents the portfolio value for both the MLTrader strategy and SPY. The right axis shows the cash balance, highlighting how the system dynamically adjusted cash holdings over time [3].

**Zoom Buttons (1m, 6m, YTD, 1y, all):** These buttons allow for interactive exploration of the graph, enabling the viewer to focus on specific time frames such as one month, six months, year-to-date, one year, or the entire period.

This graph shows, how well the MLTrader Strategy has performed and continues to perform. The strategy consistently outperformed SPY, indicating that the AI-driven system does a great job of identifying profitable trading opportunities while dynamically managing risks. Its adaptability reflects in the change of trading frequency and cash levels based on market conditions reflected in the changes of density in buy and sell markers and the respective fluctuations of the green cash line. It takes real-time market news and data to deliver timely buy and sell signals based on sentiment analysis and real-time decision-making [14].

That really underlines the outstanding performances of the strategy's qualities-ML Trader. The latter had been repeatedly on top against SPY, illustrating very well an ability of AI to disclose advantageous trades while self-momentarily managing one's risks [2]. The adaptability is evident in its adjustments of trading frequency and cash levels according to market conditions-as reflected in the varying density of buy and sell markers, and fluctuations in the green cash line. These results point is towards the possible gain of the sentiment-driven machine learning-based trading approach over the benchmark traditional passive index-tracking strategy [5].

In the graph, the green line represents cash management activity of the system with several spikes and dips. A rise in the cash correspondingly shows that the system either decreased its exposure to trades or had only negligible opportunities present in the market, possibly part of the risk management strategy [3]. These fluctuations highlight the system’s ability to dynamically allocate capital, adjusting its positions based on market conditions and sentiment analysis outcomes.

A screen shot of a computer

Description automatically generated

*Figure-3.2.2: Cash flow Tooltip*

When we move the cursor on the cash flow line, it shows a cash tooltip as shown in the figure-2, that provides a snapshot of the system's cash position at a specific moment. Cash on 18th November 2023, at 9:30 AM: -$2,257,970.83, a high negative cash balance.

From the figure-3.2.1, notice the number of green and red markers that are a testament to how often this active MLTrader Strategy rebalances and change. During periods of high market volatility, these markers are more concentrated, indicating that the system detected more opportunities to execute trades. Conversely, periods with fewer markers show the system’s conservative stance, either holding existing positions or accumulating cash to minimize risk. It gives the signature on how capable this strategy would be in respect to changeable market scenarios [47].

In this figure-3, the red tooltip in red colour showing "sell" action taken by AI-based trading system, has sold 623 shares of SPY - Exchange Traded Fund that tracks the S&P 500 index, at $366.8100 per share; the value of the transaction is $228,522.63. This trade was executed as a market order and hence was instantly filled at the best price currently available in the market. There were no associated transaction costs with this trade. The trade was initiated at 9:30 AM on September 28, 2022, when the U.S. market opened. This may reflect either the sentiment-based analytics of the system or the implementation of a set risk management rule to liquidate the position with the purpose of reducing further losses or taking gains [41].

A screen shot of a graph

Description automatically generated

*Figure-3.2.3: Sell Tooltip*

The green tooltip as shown in the figure-4, illustrates a "buy" action executed by the system. Here, the strategy bought 367 shares of SPY at a price of $408.8280 each. This lot was worth $150,039.87. The trading system placed this trade as a stop order. Whenever SPY's price reached a level where the trading system wanted to place this trade, it did so. Again, no trading costs have been considered. This is a trade that was placed on February 1, 2023, at 9:30 AM, immediately after the market opened. This purchase decision just underlines the system's ability to identify favorable market conditions and act upon them through sentiment analysis and price thresholds for possible profit.

A screen shot of a computer screen

Description automatically generated

*Figure-3.2.4: Buy Tooltip*

A noticeable spike in the MLTrader Strategy’s performance from the figure-3.2.1 is observed around 2023–2024. This period also corresponds to an increase in trading activity, as indicated by the cluster of buy and sell markers. The system appears to have effectively exploited market conditions during this timeframe, resulting in rapid growth. This, in turn, reinforces the capabilities of the system to adapt and capitalize on favourable trading opportunities p13].

In the figure-3.2.1, the orange line corresponding to SPY reflects a stable, though modest growth over the same period, which is pretty much expected for SPY, given its nature of being a passively managed index-tracking ETF for the overall performance of the market. The less volatility of its line compared to the MLTrader Strategy reflects stability at the cost of a slower but sure growth profile. This justifies the typical trade-off which one may expect while resorting to AI-driven trading strategy performance over conventional benchmark performance [51].

**Tearsheet Generated by QuantStats:**

This tearsheet has been generated by QuantStats; it gives extensive insight into the performance by the MLTrader Strategy against that of the SPY benchmark [51]. That includes various metrics, graphs, and analyses to gauge the efficiency of the strategy regarding profitability, risk management, and more. The tearsheet summarizes some key points about cumulative returns, monthly performance, drawdowns, and volatility, giving a clear contrast between the AI-driven strategy and that of the market index. This visual and numerical summary will help assess how well the system can adapt to market conditions and, therefore, be effective at yielding superior returns [29].

A screenshot of a computer

Description automatically generated

*Figure-3.2.5: MLTrader Strategy Performance Summary*

Figure 3.2.5 shows some performance indicators of the strategy of MLTrader. The strategy has an annual return of 45.5% and a total return of 529.63%, far above the standard market benchmarks [2]. Such performance comes with higher risk, as shown by the maximum drawdown of -68.94%, meaning the largest peak-to-trough loss during the period in consideration [7]. Given that it has a high drawdown, the strategy has pretty solid risk-adjusted returns. Its RoMaD stands at 0.66-a function of its capabilities of generating returns against its worst-case losses. It also saw its longest drawdown period of 768 days, representing the time it needed to recover from those high losses [16]. The strategy parameters are trading the SPY symbol with a Cash-at-Risk setting of 0.5, balancing this aggressive performance with the calculated approach to the risk management policy. These figures of merit together demonstrate the potential and the challenges in employing the MLTrader strategy [25].

A screenshot of a white and black screen

Description automatically generated

*Figure-3.2.6: MLTrader Strategy Performance metrics-I*

Figure 3.2.6 benchmark the major performance signals of SPY-an S&P 500 ETF-against the MLTrader strategy. It highlights the disparities in return, market exposure, and considerations of risk. SPY and MLTrader take a 4.21% risk-free rate as a base for returns in comparison to risk-free investments, provided by Treasury bills[4]. Their market engagement strategies are considerably different. SPY has been passively invested; thus, it reported a time in the market of 69.0%, reflecting perpetual market exposure. MLTrader's time in the market was 42.0%, implying its dynamic trading behaviour, intending to avoid adverse conditions, much of the time.

When looking at returns, MLTrader outperformed SPY significantly. Whereas SPY realized a total return of 96.05%, meaning it effectively doubled the initial investment, MLTrader earned an outstanding return of 529.63% due to its efficient AI-powered sentiment analysis and adaptive trading logic [14]. In like manner, the Compound Annual Growth Rate for MLTrader reached as high as 45.5% versus SPY's 14.71%, underlining MLTrader's superior annualized growth potential. Figure 3.2.6 highlights the analysis and underscores how much better the sophisticated trading algorithms combined with select risk management of MLTrader have been in creating significantly higher returns, compared to a traditional passive investment strategy-like SPY.



A table with numbers and letters

Description automatically generated

*Figure-3.2.7: MLTrader Strategy Performance metrics-II*

Figure 3.2.7 shows several risk and return metrics with SPY. Note that MLTrader's Sharpe Ratio - excess return delivered per unit of risk is radically more efficient for MLTrader with a score of 0.86 versus only 0.56 for SPY [3]. Similarly, RoMaD - or maximum drawdown - tells that MLTrader is much better to avoid significant losses: while SPY score is 0.44 it is 0.66 for the MLTrader Strategy. The correlation to the benchmark shows how independent MLTrader's value is at -0.17 compared to SPY's perfect correlation of 1.0. The Probabilistic Sharpe Ratio further ascertains the better risk-adjusted performance of MLTrader with 61.96% against SPY's 45.81%.

Other metrics that indicate MLTrader's higher efficiency in minimizing downside risk include Smart Sharpe at 0.82 versus 0.53, and the Sortino Ratio at 1.45 versus 0.78. Other modified metrics also support MLTrader, such as Smart Sortino and Sortino/√2, showing the robustness for the strategy in more conservative estimations. Surprisingly, MLTrader and SPY have the same Omega Ratio of 1.29, which means the equal probability that each series reaches returns above their threshold value. These results together point toward the high risk-adjusted performance of MLTrader, diversification enhanced, and its multiple outperformances over SPY



A screenshot of a graph

Description automatically generated

*Figure-3.2.8: MLTrader Strategy Performance metrics-III*

The data in Figure 3.2.8 provides information about the risk and return characteristics of MLTrader compared to SPY. The Max Drawdown metric reveals that MLTrader experienced a larger worst-case loss (-68.94%) compared to SPY (-33.68%), reflecting MLTrader’s aggressive trading approach, which comes with higher risks during downturns. The Longest DD Days metric shows that MLTrader took 768 days to recover, slightly longer than SPY’s 708 days, indicating a slower rebound from significant losses. Volatility (ann.) highlights MLTrader’s higher annualized risk (56.71%) compared to SPY (21.11%), demonstrating larger fluctuations in performance. The R² value is very low, 0.03 for both MLTrader and SPY, which means very little correlation to the SPY benchmark and hence underlines independence from market movements.  
  
Information Ratio, relating returns to risk, stands at 0.03 for both. However, the Calmar Ratio, relating return to drawdowns, stands higher for MLTrader at 0.66 compared to SPY at 0.44, reflecting better returns in relation to the risk of large losses [41]. The Skew metric also reflects a strong positive return bias of MLTrader, 2.77, against the slightly negative skew of SPY, -0.6, to indicate that MLTrader has a tendency toward extreme positive outcomes. Last but not least, the kurtosis is considerably higher for MLTrader at 43.33 versus 17.81, with more extreme outliers, which points to a higher upside with higher risk. These metrics collectively highlight the trade-offs between risk and reward in MLTrader’s strategy.

**A white background with black text

Description automatically generated**

*Figure-3.2.9: MLTrader Strategy Performance metrics-IV*

The data in Figure 3.2.9 present an analysis of the performance and risk characteristics of MLTrader versus SPY.

The Expected Daily Return shows that MLTrader (0.1%) delivers higher daily profits than SPY (0.04%), highlighting its ability to outperform over short time frames [51]. Similarly, the Expected Monthly Return of MLTrader (3.17%) significantly exceeds SPY’s (1.15%), reflecting stronger medium-term performance.

MLTrader’s Expected Yearly Return of 44.48% far surpasses SPY’s 14.41%, indicating substantial annual growth potential, nearly three times higher than the benchmark. However, this higher return comes with elevated risk. The Daily Value-at-Risk (VaR) for MLTrader is -4.74%, compared to SPY’s -1.77%, illustrating that MLTrader carries a higher likelihood of daily losses during adverse market conditions [13]. The Expected Shortfall (cVaR), which measures the average loss in extreme scenarios, is -4.74% for MLTrader versus -1.77% for SPY, highlighting the greater downside potential of MLTrader’s aggressive strategy [11].

**A screenshot of a graph

Description automatically generated**

*Figure-3.2.10: MLTrader Strategy Performance metrics-V*

To illustrate MLTrader's performance against the benchmark, we compare it with SPY (SPDR S&P 500 ETF Trust) in different time series. The services at the beginning of the month registered a monthly return of 0.89%, indicating a higher level of performance over SPY, which it surpassed by a margin of 2.06% [16.] SPY outperformed MLTrader over the 3-month (3M) and 6-month (6M) periods. This saw MLTrader returning -4.5% and -8.54%, respectively, indicating that there are some risks within the strategy with periods of inside-market conditions.

The Year-to-Date (YTD) performance demonstrates outstanding results on the part of MLTrader, because the exponential rate of growth is at 102.46% and considerably better than the 26.05% SPY has to show. Similarly, over the 1-year (1Y) range, MLTrader far outperformed SPY with an increase of 88.26%, as compared to SPY's 25.35%. For the 3-year (3Y entry) period, the annual return ended impressively at 49.08%, departing from the miserly 11.98% return yielded by SPY. Even over the really long-time frames such as the 5-year (5Y entry) and 10-year (10Y entry) periods, MLTrader sustained a consistent annualized return of 45.5%, with three times and a little more than SPY, providing a return of 14.71%. An annualized all-time return also appreciates what a solid performer MLTrader has been, as it has beaked out 45.5% versus SPY's 14.71%. These results underscore MLTrader's performance over all time frames but particularly within mid- to long-term time frames, showing its potential to present a significantly higher return than a traditional SPY model [12].

**A number of percents on a white background

Description automatically generated**

*Figure-3.2.11: MLTrader Strategy Performance metrics-V*

Figure 3.2.11 data displays a comparison below linking SPY and MLTrader strategy versus key metrics, highlighting good and worst days, months, and years. While MLTrader was much better in getting more on the short-term market opportunities-the Best Day delivering 37.47% over SPY's 9.06%-it reported an incredibly high return of 87.58% for Best Month, with SPY at 12.7% fuzzy data, expressing the claim of capitalizing on favourable market situations for exceptional growth over an extended period. Even on a longer scale, the Best Year of MLTrader showed an ever more remarkable profit return of 102.46%, versus SPY's 28.77%, illustrating its susceptibility for profitability and aggressiveness in profit-taking following a very bullish time period.

MLTrader faced the greatest losses in adverse scenarios as well. Only one day had the lowest trade-off value of -26.38%, wherein SPY's stood at -10.94%. Its worst month saw huge losses of -44.83% (SPY's -12.44%). Despite these failures, MLTrader had a good risk hedge over the long term, with its Worst Year barely recording a loss of -2.5%. This was much more commendable than SPY, which presented a loss of -18.16%. This comparison disclosed that in MLTrader, the risks were directly proportional to the potential rewards. Accelerated burstiness does not absolve the situation for shapely-shaped fund growth investors while tolerating occasional drawdowns**.**

A white paper with black text

Description automatically generated

*Figure-3.2.12: MLTrader Strategy Performance metrics-VI*

The data in Figure 3.2.12 shows the drawdown and recovery characteristics of SPY and its strategy counterpart MLTrader, allowing for comparisons. In cases of losses, SPY has been recorded to yield lower average losses owing to its lower drawdown and returning to Avg. Drawdown Recovery of 1.91%, whereas MLTrader is heavily weighed down by -13.52%. Within the given analysis, already, the recovery has been found to be much faster for 18 days for SPY compared to that for MLTrader's 79 days of recovery. Despite such a significantly high recovery due to its ability to restrain itself from loss, power of MLTrader will make any better comeback, taking its Recovery Factor to 3.77, further compared to that for SPY, only 2.33. Higher loss periods are clearly visible in the line of MLTrader, where bigger magnitude brings buyer's cheekbones wider at their higher Ulcer Index, counting 0.29 as compared with SPY having 0.09, signalling higher volatility and amount of drawdown period.

MLTrader balances this risk with a higher trade-off factor, 0.53, as opposed to SPY's 0.49. It does outline that while MLTrader is highly risky with very long recovery periods, it equally rewards higher, attracting investors who are ready to face the turmoil up front for potential long-term gains.

A graph of a number of percent

Description automatically generated with medium confidence

*Figure-3.2.13: MLTrader Strategy Performance metrics-VII*

The data in Figure 3.2.13 shows the performance comparison of the strategy versus SPY in terms of Gains and Winning Periods. This potentially maintains a very high advantage of 16.01% with the MLTrader over SPY on an average monthly gain of 4.71%, underlining the potential for higher rewards during positive months; this also brings higher risks, as shown by the strategy's much higher average monthly loss of -9.7%, more than twice the size of this loss by SPY of -3.89%.

Among positive measurements, SPY tops MLTrader in what is a count of winning days, achieving gains for 54.83% from the days covered in the analysis, while MLTrader demonstrates gains on 51.29%, showing remarkable consistency by SPY. A similar fate is bestowed on SPY should the strategypound the MLTrader in winning months (66.1% as compared to 50%) and winning quarters (75.00% as opposed to 44.44%), painting a more stable picture, in which leans slightly more toward SPY as far as the winning months and quarters are concerned. The roll rate per year on both sides comes at an equal split for wins both of the years seen, with 80% wins to the years counted in the analysis of both SPY and MLTrader, illustrating the presence of reward with higher volatility and risk in the case of MLTrader. SPY has always maintained an upward yet equal ratio to both reward vs. risk forever against its higher slippery competitor, MLTrader.

A white background with black dots

Description automatically generated

*Figure-3.2.14: MLTrader Strategy Performance metrics-VIII*

The data in Figure 3.2.14 presents MLTrader’s performance metrics in terms of risk and return relative to the market benchmark. The Beta value of -0.46 demonstrates that MLTrader moves inversely to the market, effectively reducing its exposure to market-wide risks and making it less susceptible to systemic market fluctuations. A positive Alpha of 0.6 highlights MLTrader’s ability to deliver returns exceeding the benchmark, reflecting its performance and superior strategy execution.

The negative Correlation value of -17.14% stipulates that with regard to independence from the market, and therefore possibly being a diversification tool in a portfolio, the MLTrader is weak and somewhat negatively related to the market. A Treynor Ratio of -1140.88% reflects inefficiency related to returns in terms of systemic risk (Beta). This is primarily due to the negative Beta, which can distort the conventional interpretation of this ratio. The metrics emphasize MLTrader’s ability to provide diversification and achieve market-independent returns while presenting a unique risk-return dynamic.

A table with numbers and text

Description automatically generated

*Figure-3.2.15: Comparison between SPY and ML Strategy End of Year Returns and Benchmark*

The chart in Figure-3.2.15 shows comparisons of the End-of-Year (EOY) returns achieved by the MLTrader Strategy against the SPY benchmarks from 2020 to 2024. The comparisons are done annually where number for the returns, the Multipliers, and the win/loss signs can be used for that purpose. The SPY column, as clearly described, stands for the annual returns on the market benchmark SPY. In 2020, SPY achieved a return of 17.60%; meanwhile, the MLTrader Strategy returned 98.55%, reflecting its beaming outperformance. Another one occurred in 2024, with the MLTrader reaching 102.46%-SPY standing at 26.05%-really hammering home the point about success.

The Multipliers column reveals how much SPY performed better or worse relative to the strategy. Anything higher than one signifies that it is outperformed, whereas a minus score means that they had directly contradictory returns. This means, for example, that the 5.60 multiplier from MLTrader in 2020 would outperform SPY by a factor of 5.6. On the other end, take the case of 2022: the SPY loss (-18.16%) is met with a hefty 2.61 negative multiplier to indicate that MLTrader could turn in positive returns for unfavourable market conditions.

The column representing wins visually shows whether the MLTrader strategy surpassed SPY each year. A "+" sign indicates a win, and a "-" indicates the underperformance. MLTrader outperformed SPY in 2020, 2022, and 2024, as indicated by the "+" symbols, but underperformed in 2021 and 2023. This table shows how, in the years like 2020 and 2024, MLTrader can significantly outperform and perform resiliently in bad markets like 2022, further justifying the effectiveness of the strategy. The Multiplier and Won metrics underline its consistent potential for superior returns compared to SPY.

A table with numbers and numbers

Description automatically generated

*Figure-3.2.16: Worst 10 Drawdowns*

The "Worst 10 Drawdowns" table within Figure-3.2.16 illustrates how the MLTrader Strategy suffered its most significant losses. A drawdown is a proportion of losses since the peak value before a recovery began. It is an understandable representation in terms of the risks an individual's strategy will undergo. The largest drawdown has been at -68.94% from high to low, from 2022-08-16 to 2024-03-07, and it had a very long recovery period of 570 days. Some other potentially unnerving drawdowns: -38.16%, with a recovery after 56 days through mid-2022; -32.14%, on a recovery which took 768 days from the start of 2020. These episodes have shown that during the real drawdowns, the strategy can stay in a very sensitive position and requires more aggressive time to redeem it.

By the way, the table shows some examples of resilience where, in a relatively short time, as for an example, the -16.81% fall of a drawdown in the year 2022 rebounded in just 2 days; and in 2024 it fell by -7.93% in 13 days. These reduced recoveries actually show that under certain conditions, the strategy works well and profits from shifting market conditions. In fact, those figures in bold face are quite telling, it calls for an appropriate analysis of the magnitude and length of the drawdowns to understand how the overall behaviour has changed due to a variation in the risk profile for the strategy. In spite of the severe drawdowns, the strategy's recovery in recovery over a substantially long time is a clear marker of its resilience—thus offering an interesting choice for those investors who can withstand any short-term volatility with long-term gain ahead of them [15].

**Graphical Results:**

A graph of a stock market

Description automatically generated

*Figure-17: Comparison between Cumulative Returns and Benchmarks*

In the line graph, Figure-3.2.17 shows the comparison in cumulative returns between the MLTrader Strategy and the SPY benchmark from the Exchange Traded Fund (ETF) over the analyzed period.

The MLTrader Strategy clearly outperforms SPY as of the beginning of 2022. While the cumulative gain by the end of 2024 has surpassed the 600%-mark, SPY had been growing gradually, with no dramatic changes in its performance-a typical picture for a passive investment strategy. It is clear from this that the MLTrader Strategy is taking profit out of the market because it dynamically leverages opportunities and changes strategy, while SPY is just standing still, like a rock, unchanging. The massive spread between the two strategies clearly showcases that the MLTrader Strategy gives an investor more reality and possibilities to seek and give greater returns compared to a traditional index-based investment' for aggressive growth[27].

A graph of a graph showing the number of stocks

Description automatically generated with medium confidence

*Figure-3.2.18: Comparison between Cumulative Returns and Benchmarks (Log Scaled)*

Line graph (from the figure-3.2.18) for comparative performance displayed between the logarithmic plotting of the MLTrader Strategy and that of SPY (yellow line). The scaling is logarithmic for the sake of displaying proportional growth over time, thereby enabling one to check on percent growth instead of absolute numbers. This shows that the MLTrader Strategy outperforms and lives up to outstanding performance from 2021 onwards in an exponential climb, culminating at a much higher cumulative return by 2024 in respect to SPY.

The SPY shows a steady, linear progression due to the relative reference of the market benchmark. Hence, the logarithmic scales serve as examples of capturing the growing results of SPY on both the stable side and the lesser side compared to the volatile and dynamic returns of the MLTrader Strategy. This also establishes how good the strategy is at idle stock performance in order to deliver higher but risk-filled returns.

A graph of a graph showing the value of a stock market

Description automatically generated

*Figure-3.2.19: Comparison between Cumulative Returns and Benchmarks (Volatility Matched)*

The comparison of MLTrader Strategy (blue line) and SPY (yellow line) is shown in Figure-3.2.19, with returns adjusted for using volatility for a risk-adjusted analysis. This is in consideration of highlighting the fact that returns of MLTrader Strategy are significant in risk-adjusted return contexts, making any comparison more valid .

In the bar chart above, MLTrader Strategy gives a superior performance compared to relevant index ETF, SPY. Cumulative returns are increasing at a much steeper rate after 2023 up to 2024. Unlike the MLTrader Strategy that shows good adaptation to prevailing market conditions and prompt decision-making, SPY remains quite stabile and has a steady increase over time in accordance with market trends moving upward. This further reinforces the idea that the high returns from the strategy do not necessarily come with added risk; these are made by well-managed trades. Also, the strong performance is proved under the complex and competitive financial environment [42].

A graph with numbers and a line

Description automatically generated

*Figure-3.2.20: Comparison between EOY Returns and Benchmark*

The line graph above from Figure-3.2.20 represents the End-Of-Year (EOY) returns of the MLTrader Strategy (in blue) against the benchmark SPY (in yellow) for different years running from 2020-2024. One can count on this visualization for a year-by-year comparison of the strategy against the market benchmark: therefore, by giving an edge for adaptation best-suited to the changing market-environments.

2020: That year, the MLTrader Strategy strongly outperformed the SPY, turning a fabulous return of 98.55% by comparison to the latter’s 17.60%. This was the evidence of an excellent performance in favorable conditions with respect to capitalization.

2021: In the given period, the SPY excelled against MLTrader, delivering a solid 28.77% return and a relatively smaller loss of -2.50% for the strategy. This could project either the poorly conducive performance of the MLTrader or lack of exploitable opportunities through its instruments.

2022: The declining SPY figures, on the other hand, excited MLTrader to reach 47.47%, indicating how resilient MLTrader is and the excellent predicts of the bear market for profit making.

2023: Ultimately, it was a matured market to a score of 25.50% that gave the greater profit overall against the return of only 8.94% by MLTrader, which means the year was less aggressive in performance for the strategy.

2024: The MLTrader Strategy thus goes for other peaks with a 102.46% return against SPY's 26.05 percentages. There is so much evidence here that the MLTrader Strategy is simply superb in understanding trends for profits.

An arbitrary threshold of 45%-approximately, indicated by the red dashed line on this chart- would thereafter be considered average performance or a target. Hence, this chart spells out the extant potential for superior long-term returns of the MLTrader Strategy, despite probable occasional underperformance, leading to an argument for its being a very robust and high-potential trading strategy [56].

A graph of a distribution of monthly returns

Description automatically generated

*Figure-3.2.21: Distribution of Monthly Returns*

The figure of 3.2.21 above shows the distribution of monthly returns for the MLTrader Strategy in blue, while yellow denotes the performance of SPY, providing us with a contrast of how they have been differentiating themselves on return and risk.

The point of the SPY distribution, at least the narrowed range because of concentrated returns near 0%, denotes a very stable frame of moderate monthly returns with low particular or monthly volatility. SPY is the slow and steady benchmark investment type. The MLTrader Strategy with its distribution at the extended edge of SPY's shows an even wider picture of return with more human entries both higher and lower than SPY brandishing in and out with large gain to give rise to as high as 80% followed by bigger negative returns continuing down to -40% indicating exaggerated trading behavior for the quest for a high risk at high returns.

Considering the overlap in the center of both distributions, the similarity of moderate returns is somewhat uniform, but the spread of the MLTrader Strategy in return closely indicates that the system can capture market movement on a larger scale. While we identify a higher level of risk in this strategy, its wide range of higher profitabilities suggests it can enlarge the peak earnings during a month compared to SPY's steadier results. The distribution underscores the tradeoff notion between the increased risk involved and the hopes for higher returns from the other angle.

A graph showing a line of blue lines

Description automatically generated with medium confidence

*Figure-3.2.22: Distribution of Monthly Returns*

Graph of daily return in the figure 3.2.22, denotes day-wise percentage shifts in the performance of the MLTrader Strategy, presenting itself with volatility and flexibility in coping with market situations. It is capable of spiking return percentage in both positive as well as negative directions during intensity of activities in the stock market due to it reacting to any impacting news or significant events.

There are periods when daily return stable around 0% connoting a period of calm or stability, during which the strategy is forced not to trade actively or mostly faces very minimal movement in the market. Greater heights of 40% of return from the strategy stand for its exposure to seizing profitable opportunities; correlatively, any downward movement slicing through -20% losses catches exposure to the risky attributes attached to an aggressive trading ability. The characteristic that the graph conveys of the MLTrader Strategy is its dynamic and incidental nature. The strategy does not hesitate to active participation at a time of daily market movement, leaving it for either a high return to be obtained or risk imbedded in a complete volatile financial environment.

A graph with numbers and lines

Description automatically generated

*Figure-3.2.23: Rolling Beta to Benchmark*

From the figure-3.2.23, indicates representation under Rolling Beta to the Benchmark is projecting a sight of the tendency of the MLTrader Strategy toward the benchmark market fluctuation (in this case, SPY) over two different rolling periods: for six months (highlighted by the blue curve) and then for 12 months (highlighted by the grey one). Beta is the level of the correlation of the strategy performance with the benchmark. A beta near 1 signifies the strategy's moves alongside the market, those exceeding 1 say that the market effects on the sharpy straighten up. The reverse of this shows negative beta values-for example, during the downturns when the strategy produces better returns.

This exhibits a trend that captures, within this changing pattern of beta-value averages across time. The example is beta spikes in 2024, which indicates a very high resemblance to SPY and suggests that in general, during positive market times, this strategy went off really well. The period with a negative-like beta, possibly in mid-2022, underlines a better hedging and profitability of such strategy in case of bad markets. The red dotted line at the 0 value indicates a neutral level at which the strategy neither follows the trends of nor acts against the market.The chart illustrates the dynamic nature of the MLTrader Strategy. The chart urges the viewer to think of the MLTrader Strategy as right on top of risk management but varying between alignment with or counter- to market moods, subject to the then-prevailing conditions.[57]

A graph of a graph with numbers and a line

Description automatically generated

*Figure-3.2.24: Rolling Volatility (6-Months)*

The Rolling Volatility (6 months) Review in Figure-3.2.24 takes an in-depth look at the comparison between the 6-month rolling volatilities of the MLTrader Strategy (blue line) and those of the SPY benchmark (yellow line). Volatility, which takes into account the degree of return fluctuations, stands as a key performance indicator and a measure of risk or uncertainty in performance. Indeed, the MLTrader Strategy saw various sections of increased volatilities, especially during the time around 2022 and then 2024, which may be associated with strong trading opportunities, and its reaction to changing market conditions to present a continuously changing type of risk profile. SPY, on the other hand, showed very mild and steady volatility, consistent with the considerations of broad market indexing.

The red dashed line, shown by the 0.5 value, indicates the key level of volatility. It is a perfect time to take courageous action when the MLTrader Strategy's volatility reaches this level or when it becomes highly reactive to market movements. It may therefore follow in SPY's footsteps along a shallower, undulating path with less volatility. This variance now serves as a good example of the asset class's versatility in showing how the MLTrader Strategy fits for opportunity capture with dynamic risk management in multiple market conditions.

A graph with numbers and lines

Description automatically generated

*Figure-3.2.25: Rolling Sharpe (6-Months)*

The Rolling Sharpe (6 months) plot in Figure-3.2.25 depicts six monthly variations in the Sharpe Ratio of the MLTrader Strategy. The Sharpe Ratio is a widely accepted risk-adjusted benchmark that compares excess returns of a portfolio with respect to a risk-free rate against its volatility. The dotted red line stipulates 0, which is a neutral point, dividing the good Sharpe Ratio periods from the poor Sharpe Ratio periods.

Those times when the strategy was outperforming were represented by facing above levels of the surrounding bands. At such times, the strategy was generating returns that were completely commensurate with the risks taken, and risking absolutely had already been very well rewarded, during late 2020 and mid-2021, such that these were shining moments in risk-adjusted performance. From the other angle, the recesses from the below levels of the bands, like early 2022 and early 2025, mean that for a given level of risk, the returns from the strategy were lousy. These are the challenging times when the market went against the strategy, or for whatever reason is trading unsoundly. Peaks were those times when the market tried to reward the strategy with good performances. The troughs are the warning sign of the potential improvement side of risk and return coordination when the conditions were not as good. This visualization is, therefore, an interesting performance tracker over time.

The swings on the chart illustrate the adaptability of the strategy to changing circumstances. Peaks were those times when the market tried to reward the strategy with good performances. The troughs are the warning sign of the potential improvement side of risk and return coordination when the conditions were not as good. This visualization is, therefore, an interesting performance tracker over time.

A graph of a graph showing the growth of a stock market

Description automatically generated with medium confidence

*Figure-3.2.26: Rolling Sortino (6-Months)*

From the figure-3.2.26, the Rolling Sortino (6-Months), which exhibits the six-month variations in the Sortino Ratio for the MLTrader Strategy. The Sortino Ratio measures risk-adjusted returns but focuses on downside risk—negative departures from the target or risk-free return. Values above the zero line indicate that the strategy has provided positive returns against its downside risk, and values below the zero shows periods of underperformance in terms of negative risk management.

The emergence of high peaks, like the surge in mid-2021, is exemplary about the risk-adjusted performance by the strategy because it capitalized on the market conditions prevailing at that time and resulted in minimum further downside losses. Periods of heightened downside risk or, rather, where returns could not match negative market movements portend dark weeks, made worse when one or two of these instances badly coincide with market conditions. The graph illustrates how, under changing circumstances, varying levels of loss-prevention have been maintained by the strategy. Mostly on the back of positive Sortino Ratios throughout the considered period, the strategy has-with few exceptions-managed to consistently mitigate downside risks and deliver favorable returns. All these are evidence of the strategy miserably working in Favor of its prime goal, i.e., maximum downside protection, in light of the risk-reward balance, even under changing market scenarios.

A graph showing a graph of a graph

Description automatically generated

*Figure-3.2.27: Strategy – Worst 5 Drawdown Periods*

Strategy-Worst 5 Drawdown Periods shown in figure-3.2.27 make an indication of the worst and deep dive (drawdowns reach less than -10%) performance, shaded in red, in the history of the portfolios. Drawdown is measured as the percentage of depreciation from the peak value of the portfolio at the trough. They can be the period in history where nearly perfect storm times are hitting the open market enough so that the strategy could approach a new, better position in severe conditions of the market.

Specific periods, particularly 2022-2023, are described with greater illustration from the standpoint of the strategy's long and significant drawdowns and seeming effort to recover to prior peaks. This underscores the opportunity for enhancing risk management to such an end that would provide better performance on the downside. These strategies provide an upward trajectory by showing recovery and growth after each dip. The visualization shows the importance of balancing large-scale risk-taking with effective management to facilitate long-term success.

A graph showing a line of water

Description automatically generated with medium confidence

*Figure-3.2.28: Underwater Plot*

The figure-3.2.28, presents a glide through the drawdowns of the strategy over time; they represent the percentage declines from the peak value of the portfolio. The trade-offs, shown by blue shading having prices below zero, indicate levels of drawdown of previous peak values; these are darker and more expressive of the amount of the strategy's risk and its recovering performance.

There are notable deep falls in 2022 and 2023 that mark substantial drawdowns and are monitored by using the red dashed line drawn for performance below this target (e.g., -20%). The picture shows how, in most cases, the strategy returned and surpassed its previous peaks, a point that emphasizes the strategy's resilience. This graph is essential to understanding the continuity and efficiency of the capacitance of the strategy in managing risks and gives a hint about its loss mechanism and long-term growth.

A chart with numbers and a number of months

Description automatically generated with medium confidence

*Figure-3.2.29: Strategy - Monthly Returns (%)*

From figure 3.2.29, the Strategy Monthly Returns (%), giving the month-by-month breakdown of the performance by the strategy from 2020 to 2024. Each colour-blocked cell presents a % return for the month, with a color gradient indicating the performance: shades of green indicate the good returns, orange and red denotes the negative returns. The intensity of the colours corresponds to the magnitude of the returns.

March 2020 (68.68%) and June 2022 (87.58%) are cheered as the most profitable periods of strategy, not only in deep green. November 2023 (-44.83%) is uniquely conspicuous in bright orange as the juiceless one. With application of some seasonally or cyclically driven effects, it yields further proofs of months that are the strongest or the weakest in terms of performance. The heatmap also flashes an insight into the overall volatility of the strategy, focusing on how constant variation can point out different years' extreme fluctuations. This analysis serves in

A graph of a diagram

Description automatically generated with medium confidence

*Figure-3.2.30: Strategy - Return Quantiles*

The Strategy - Return Quantiles box plot in Figure-3.2.30, shows the distribution of the strategy's returns of several time intervals as daily, weekly, monthly, quarterly, and yearly. The figure-3.2.30, indicates the statistical metrics for each interval. The interquartile range(IQR), represents the box that captures the central 50% of returns, and the horizontal line within the box that indicating the median return. Whiskers extends till the values within 1.5 times the IQR, with individual points outside this range identified as outliers.

This plot shows that the longer the time period, the more spread the returns. Similarly, daily and weekly returns have relatively narrow distributions, signifying consistent performance over a while. The yearly returns on the other hand, shows strong variability and can result in huge returns higher than 100%, as shown by the outliers. This trend underscores the compounding effect and the strategy’s capacity to achieve large returns over extended periods. The visualization is a powerful tool for assessing the risk-return dynamics of the strategy, emphasizing its behavior across various investment horizons [59].

* 1. **Development Insights:**

The insights from development may point to ways to improve the AI trading system's ability to perform and adapt. Included is a list of efforts aimed at enhancing the accuracy of decision-making and risk management, as well as providing the effective tools that actually render reliable results on a dynamic platform in the markets. The main idea behind these development insights is to better equip armed users for use of the system by addressing crucial challenges. The following are some insights to develop in the system:

Improved Risk Management: The system has faced huge losses during market downtrends and therefore needs stronger risk management. This can involve some technical means, such as the implementation of dynamic-level stop loss and take-profit points that change automatically with the changes in market conditions. A side product of diversification of trading pairings with different types of assets will result in minimalizing risk while increasing exposure of the full trade [7].

More Versatile Sentiment Analysis: Lots of decisions are based on sentiment analysis. However, there are times when this tool fails to provide accurate information. Point tuning of the FinBERT model should be done along with broadening the variety of financial data to produce results that can be predicatively consistent. The addition of some financial information will help outpace the sentiment scores with decisions [33].

Smarter Trading Frequency: Overtrading could incur unnecessary costs and amplify risks. The system can use historical data to understand when to actively trade while avoiding trading in low-confidence periods, thereby reducing unnecessary risks [3].

Ability to Adapt in Different Market Conditions: The system is simply not good in very volatile markets. More trading tools next to sentiment analysis should better equip it. Machine-learning models that learn from changing market patterns might render it adaptable [13].

Another pretty useful idea is graphic representation of the best visual display in the performance of the system should include a real-time dashboard featuring important metrics like profits, risks, position sizing, live trades executed, etc [19].

# **Conclusion:**

## **Summary of Findings and Insights:**

The trading system sustained by artificial intelligence dictates a terrain in altering financial decisions by means of implementing sophisticated machine learning techniques involving FinBERT to analyze market sentiment and Alpaca API, thereby utilizing algorithms developed in the capacity of trade strategy [33]. During the backtesting period of 2020-2024, the system achieved a cumulative return of 529.63% and an annual return of 45.5%. Risk-adjusted performance suggests that the strategy, is good on a global scale, with the metrics that suggest it has the potential to maximize returns, with the Sharpe Ratio at 0.86 and Sortino Ratio at 1.45 for 68.3% off and 3.5-streak [7].

The system's feature on making a profit from sentiment-driven market opportunities, engaging in trade management, and outperforming traditional benchmarks; however, the challenges involving this system are paramount: large drawdowns, with the maximum at -68.94%, prolonged recovery from potential opportunities. Working, with sentiment analysis drew more attention efficiently yet revealed a need for more data cash and improved modelling accuracy so as to limit errors [13].

The results of this analysis particularly advocate for enhancing risk management, such as dynamic stop loss, take profit, and improved adaptability for extreme market conditions, all with the incorporation of diverse data sources. Further optimization could be on sentiment analysis models, as well as Vis Tools, for purposes of bettering the system in which one is generated to achieve consistent profitability in the market.

## **Future Direction**

Integration of Additional Data Sources: The integration of various datasets such as macroeconomic indicators, geopolitical events, and real-time global financial news will supply continuous analysis. The system's predictive ability in market trends, coupled with its capacity to detect opportunities, will improve its overall effectiveness [18].

Advanced Algorithmic Enhancements: The adoption of machine learning models like reinforcement learning or ensemble methods would affect decision-making, employing algorithms that dynamically alter themselves according to developing changes in the market and will possibly result in more stable trading strategies [23].

Improvement in Sentiment Analysis Models: Improvement via hybrid models including FinBERT and any other NLP framework will improve accuracy in the sentiment analysis component. Custom training on different datasets using annotated polarity lexicons would polish the predictions [14].

Dynamic Risk Management: Executing a couple of adaptive risk controls that could possibly be merged into stop-loss and take-profit mechanisms with variable triggers reflecting market conditions shall help diminish exposure during periods of loss. Changing its trading among different types of assets will further reduce the earlier risks [44].

Enhanced User Interfaces: Building intuitive dashboards and real-time visualization tools will better users' experience. Product features like customizable metrics and predictive analytics will allow users to execute well-informed real-time strategy changes [19].

These future directions are aimed at ensuring a reliable, scalable, and user-adopted system that is a success going forward in varied financial markets of constant transformation.

# **References:**

1. A. B., Saputro, Prasetijo, T. A., Windasari, I. P., & Windarto, Y. E. (2017). Buy/sell signal detection in stock trading with bollinger bands and parabolic SAR: With web application for proofing trading strategy. *Essay*, 41–44 <https://doi.org/10.1109/icitacee.2017.8257672>
2. Abrol, S., Chesir, B., Mehta, N., & Ziegler, R. (2016b). High Frequency Trading and US Stock Market Microstructure: A Study of Interactions betweenComplexities, Risks and Strategies Residing in U.S. Equity Market Microstructure. *Financial Markets Institutions and Instruments*, *25*(2), 107–165. <https://doi.org/10.1111/fmii.12068>
3. Adams, J. J. (2023b). Moderating noise-driven macroeconomic fluctuations under dispersed information. *Journal of Economic Dynamics and Control*, *156*, 104752. <https://doi.org/10.1016/j.jedc.2023.104752>
4. Akin, I., & Akin, M. (2024). Behavioral finance impacts on US stock market volatility: an analysis of market anomalies. *Behavioural Public Policy*, 1–25. <https://doi.org/10.1017/bpp.2024.13>
5. Alibašić, H. (2023b). Developing an ethical framework for responsible artificial intelligence (AI) and machine learning (ML) applications in cryptocurrency trading: A Consequentialism Ethics analysis. *FinTech*, *2*(3), 430–443. <https://doi.org/10.3390/fintech2030024>
6. *Algorithmic trading*. (2011b, November 1). IEEE Journals & Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/5696713>
7. Andersen, T. G., Bollerslev, T., Christoffersen, P. F., & Diebold, F. X. (2013b). Financial risk measurement for financial risk management. In *Handbook of the economics of finance* (pp. 1127–1220). <https://doi.org/10.1016/b978-0-44-459406-8.00017-2>
8. Ball, R. (2009b). The global financial crisis and the efficient market hypothesis: What have we learned? *Journal of Applied Corporate Finance*, *21*(4), 8–16. <https://doi.org/10.1111/j.1745-6622.2009.00246.x>
9. Betancourt, R., & Chen, S. (2019b). Date and time. In *Apress eBooks* (pp. 295–372). <https://doi.org/10.1007/978-1-4842-5001-3_7>
10. Beverelli, C., Keck, A., Larch, M., & Yotov, Y. V. (2018). Institutions, Trade and Development: A Quantitative analysis. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3167749>
11. Biecek, P., & Burzykowski, T. (2021b). Explanatory model analysis. In *Chapman and Hall/CRC eBooks*. <https://doi.org/10.1201/9780429027192>
12. *Brown, S. (2021) Machine Learning explained. - References - Scientific Research Publishing*. (n.d.-b). <https://www.scirp.org/reference/referencespapers?referenceid=3759248>
13. Cao, L. (2022b). AI in Finance: Challenges, Techniques, and Opportunities. *ACM Computing Surveys*, *55*(3), 1–38. <https://doi.org/10.1145/3502289>
14. Carter, C. (2022b, May 1). *Stock price prediction using sentiment analysis and LSTM*. <http://hdl.handle.net/20.500.12648/7334>
15. Carter, D. (2018b). How real is the impact of artificial intelligence? The business information survey 2018. *Business Information Review*, *35*(3), 99–115. <https://doi.org/10.1177/0266382118790150>
16. Carè, R., & Cumming, D. (2024b). Technology and automation in financial trading: A bibliometric review. *Research in International Business and Finance*, *71*, 102471. <https://doi.org/10.1016/j.ribaf.2024.102471>
17. Chen, Y., Wan, Z., Zhuang, Y., Liu, N., Lo, D., & Yang, X. (2024b). Understanding the OSS communities of deep learning frameworks: A comparative case study of PyTorch and TensorFlow. *ACM Transactions on Software Engineering and Methodology*. <https://doi.org/10.1145/3705303>
18. Chowdhury, A., Uddin, M., & Anderson, K. (2021b). Trading behaviour and market sentiment: Firm-level evidence from an emerging Islamic market. *Global Finance Journal*, *53*, 100621. <https://doi.org/10.1016/j.gfj.2021.100621>
19. Chy, M. K. H., & Buadi, O. N. (2023b). Role of data visualization in Finance. *American Journal of Industrial and Business Management*, *13*(08), 841–856. <https://doi.org/10.4236/ajibm.2023.138047>
20. Ciuriak, D., & Artyushina, A. (2023). Trading AI: machine knowledge capital and the trading system. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4507114>
21. Cooper, R., Ong, M., & Van Vliet, B. (2015b). Multi-scale capability: A better approach to performance measurement for algorithmic trading. *Algorithmic Finance*, *4*(1–2), 53–68. <https://doi.org/10.3233/af-150043>
22. Cui, Y., Yang, Z., & Yao, X. (2023b, April 17). *Efficient and effective text encoding for Chinese LLAMA and Alpaca*. arXiv.org. <https://arxiv.org/abs/2304.08177>
23. Davenport, T., Guha, A., Grewal, D., & Bressgott, T. (2019b). How artificial intelligence will change the future of marketing. *Journal of the Academy of Marketing Science*, *48*(1), 24–42. <https://doi.org/10.1007/s11747-019-00696-0>
24. Downen, T., Kim, S., & Lee, L. (2023). Algorithm aversion, emotions, and investor reaction: Does disclosing the use of AI influence investment decisions? *International Journal of Accounting Information Systems*, *52*, 100664. <https://doi.org/10.1016/j.accinf.2023.100664>
25. Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P. V., Janssen, M., Jones, P., Kar, A. K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., . . . Williams, M. D. (2019). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, *57*, 101994. <https://doi.org/10.1016/j.ijinfomgt.2019.08.002>
26. Goldstein, I., Spatt, C. S., & Ye, M. (2024b). The next chapter of big data in finance. *Review of Financial Studies*. <https://doi.org/10.1093/rfs/hhae083>
27. Green, H., & Pearson, M. (2002). Artificial intelligence in financial markets. *Essay*, *2*, 839–844. <https://doi.org/10.1109/icnn.1995.487527>
28. Gupta, S., Kumar, V., Ahalawat, K., & Sagar, V. (2024b). Artificial intelligence role in stock market: A comprehensive analysis. In *CRC Press eBooks* (pp. 630–635). <https://doi.org/10.1201/9781003559085-108>
29. Gunawan, K., & Ikrimach, I. (2024). Implementation of Python-Based TOPSIS method for best stock selection analysis using Yahoo Finance. *JURNAL TEKNOLOGI DAN OPEN SOURCE*, *7*(2), 125–137. <https://doi.org/10.36378/jtos.v7i2.3873>
30. Harit, A., Sun, Z., Yu, J., & Moubayed, N. A. (2024b, August 31). *Breaking Down Financial News Impact: A Novel AI Approach with Geometric Hypergraphs*. arXiv.org. <https://arxiv.org/abs/2409.00438>
31. Harris, M. (2022b, August 21). *An even easier introduction to CUDA | NVIDIA Technical blog*. NVIDIA Technical Blog. <https://developer.nvidia.com/blog/even-easier-introduction-cuda/>
32. Hausner, J. F., & Van Vuuren, G. (2021b). Portfolio performance under tracking error and benchmark volatility constraints. *Journal of Economics Finance and Administrative Science*, *26*(51), 94–111. <https://doi.org/10.1108/jefas-06-2019-0099>
33. Huang, A. H., Wang, H., & Yang, Y. (2022c). FinBERT: A Large Language Model for Extracting Information from Financial Text\*. *Contemporary Accounting Research*, *40*(2), 806–841. <https://doi.org/10.1111/1911-3846.12832>
34. Jeong, D., Park, C., Kang, T., Shin, K., Choi, S., & Hwang, J. (2023b). Measuring ethics level of technological topics using phylogenetic tree. *Technology Analysis and Strategic Management*, *36*(11), 3417–3430. <https://doi.org/10.1080/09537325.2023.2209214>
35. Jeong, D., Park, C., Kang, T., Shin, K., Choi, S., & Hwang, J. (2023c). Measuring ethics level of technological topics using phylogenetic tree. *Technology Analysis and Strategic Management*, *36*(11), 3417–3430. <https://doi.org/10.1080/09537325.2023.2209214>
36. Kavitha, D., Avala, S. K., Bohra, S., & Srinivas, V. R. (2024b, June 15). *Real Time Sentiment -Driven Portfolio Optimization: a Machine Learning approach. | EBSCOHost*. <https://openurl.ebsco.com/EPDB%3Agcd%3A6%3A5907254/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A181715105&crl=c&link_origin=scholar.google.com>
37. Kim, Y., Jeong, S. R., & Ghani, I. (2014). Text opinion mining to analyze news for stock market prediction. \*Int. J. Advance. Soft Comput. Appl, 6\*(1), 2074-8523.
38. Ko, S., Cho, I., Afzal, S., Yau, C., Chae, J., Malik, A., Beck, K., Jang, Y., Ribarsky, W., & Ebert, D. S. (2016). A survey on Visual Analysis Approaches for Financial data. *Computer Graphics Forum*, *35*(3), 599–617. <https://doi.org/10.1111/cgf.12931>
39. Kumar, D., Meghwani, S. S., & Thakur, M. (2016b). Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. *Journal of Computational Science*, *17*, 1–13. <https://doi.org/10.1016/j.jocs.2016.07.006>
40. Lawal, A. I., Nwanji, T. I., Asaleye, A., & Ahmed, V. (2016). Economic growth, financial development and trade openness in Nigeria: An application of the ARDL bound testing approach. *Cogent Economics & Finance*, *4*(1), 1258810. <https://doi.org/10.1080/23322039.2016.1258810>
41. Li, N., Liang, X., Li, X., Wang, C., & Wu, D. D. (2009b). Network environment and financial risk using machine learning and sentiment analysis. *Human and Ecological Risk Assessment an International Journal*, *15*(2), 227–252. <https://doi.org/10.1080/10807030902761056>
42. Li, Z., Chard, R., Ward, L., Chard, K., Skluzacek, T. J., Babuji, Y., Woodard, A., Tuecke, S., Blaiszik, B., Franklin, M. J., & Foster, I. (2020b). DLHub: Simplifying publication, discovery, and use of machine learning models in science. *Journal of Parallel and Distributed Computing*, *147*, 64–76. <https://doi.org/10.1016/j.jpdc.2020.08.006>
43. Lillo, F., & Mantegna, R. N. (2000b). Variety and volatility in financial markets. *Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, *62*(5), 6126–6134. <https://doi.org/10.1103/physreve.62.6126>
44. Liu, H., Huang, S., Wang, P., & Li, Z. (2021b). A review of data mining methods in financial markets. *Data Science in Finance and Economics*, *1*(4), 362–392. <https://doi.org/10.3934/dsfe.2021020>
45. Liu, Z., Zhang, K., & Miao, D. (2024b). Evolving Financial Markets: The impact and efficiency of AI-Driven trading strategies. In *IFIP advances in information and communication technology* (pp. 301–312). <https://doi.org/10.1007/978-3-031-71253-1_22>
46. Mehare, H. B., Anilkumar, J. P., & Usmani, N. A. (2023b). The Python programming language. In *Springer eBooks* (pp. 27–60). <https://doi.org/10.1007/978-3-031-22206-1_2>
47. Mehra, S., Aggarwal, S., & Mitra, P. (2024b). Predictive analytics and AI-Driven strategies for enhanced cash flow forecasting. In *Lecture notes in networks and systems* (pp. 296–315). <https://doi.org/10.1007/978-3-031-66329-1_21>
48. Meylani, R. (2024). A Critical Glance at Adaptive Learning Systems using Artificial intelligence: A systematic review and qualitative synthesis of contemporary research literature. *Batı Anadolu EğItim Bilimleri Dergisi*. <https://doi.org/10.51460/baebd.1525452>
49. *Microsoft Visual Basic 2013 Step by step*. (n.d.-b). Google Books. <https://books.google.hu/books?hl=en&lr=&id=z7JCAwAAQBAJ&oi=fnd&pg=PR3&dq=Visual+Studio+Code+Documentation+Microsoft.++&ots=Ey8SFvd3>
50. *Natural language processing and text mining - ProQuest*. (n.d.-b). <https://www.proquest.com/scholarly-journals/natural-language-processing-text-mining/docview/3081429596/se-2>
51. Nuti, G., Mirghaemi, M., Treleaven, P., & Yingsaeree, C. (2011). Algorithmic trading. *Computer*, *44*(11), 61–69. <https://doi.org/10.1109/mc.2011.31>
52. Petry, J. (2020b). From national marketplaces to global providers of financial infrastructures: exchanges, infrastructures and structural power in global finance. *New Political Economy*, *26*(4), 574–597. <https://doi.org/10.1080/13563467.2020.1782368>
53. Prasetijo, A. B., Saputro, T. A., Windasari, I. P., & Windarto, Y. E. (2017b). Buy/sell signal detection in stock trading with bollinger bands and parabolic SAR: With web application for proofing trading strategy. *Essay*, 41–44. <https://doi.org/10.1109/icitacee.2017.8257672>
54. Rajput, A. (2019b). Natural language processing, sentiment analysis, and clinical analytics. In *Elsevier eBooks* (pp. 79–97). <https://doi.org/10.1016/b978-0-12-819043-2.00003-4>
55. Reinertsen, T., & Håkedal, B. (2021). Backtesting trading strategies on the Oslo Stock Exchange: Can a non-professional investor beat the market with financial ratios?
56. Schmidt-Kessen, M. J., Eenmaa, H., & Mitre, M. (2022b). Machines that make and keep promises - Lessons for contract automation from algorithmic trading on financial markets. *Computer Law & Security Review*, *46*, 105717. <https://doi.org/10.1016/j.clsr.2022.105717>
57. School, G. U. (2021b, July 8). *FINTECH COMPANIES: INNOVATION, ALGORITHMS AND CUSTOMER CENTRIC PERSPECTIVE - A cross-sectional study on algorithmic trading in the Fintech industry*. <http://hdl.handle.net/2077/69156>
58. Seong, N., & Nam, K. (2020b). Predicting stock movements based on financial news with segmentation. *Expert Systems With Applications*, *164*, 113988. <https://doi.org/10.1016/j.eswa.2020.113988>
59. Torka, S., & Albayrak, S. (2024b, December 14). *ALPACA -- Adaptive Learning Pipeline for Comprehensive AI*. arXiv.org. <https://arxiv.org/abs/2412.1095>