

Risk-Adjusted Performance of Random Forest Models in High-Frequency Trading

Akash Deep^{*1}, Abootaleb Shirvani², Chris Monico¹, Svetlozar Rachev¹, and Frank J. Fabozzi³

¹Department of Mathematics and Statistics, Texas Tech University, USA

²Department of Mathematical Sciences, Kean University, USA

³Johns Hopkins Carey Business School, Johns Hopkins University, USA

March 4, 2025

Abstract

Because of the theoretical challenges posed by the Efficient Market Hypothesis to technical analysis, the effectiveness of technical indicators in high-frequency trading remains inadequately explored, particularly at the minute-level frequency, where effects of the microstructure of the market dominate. This study evaluates the integration of traditional technical indicators with random forest regression models using minute-level SPY data, analyzing 13 distinct model configurations. Our empirical results reveal a stark contrast between in-sample and out-of-sample performance, with R^2 values deteriorating from 0.749–0.812 during training to negative values in testing. A feature importance analysis demonstrates that primary price-based features dominate the predictions made by the model, accounting for over 60% of the importance, while established technical indicators, such as RSI and Bollinger Bands, account for only 14%–15%. Although the indicator-enhanced models achieved superior risk-adjusted metrics, with Rachev ratios between 0.919 and 0.961, they consistently underperformed a simple buy-and-hold strategy, generating returns ranging from -2.4% to -3.9%. These findings challenge conventional assumptions about the usefulness of technical indicators in algorithmic trading, suggesting that in high-frequency contexts, they may be more relevant to risk management rather than to predicting returns. For practitioners and researchers, our findings indicate that successful high-frequency trading strategies should focus on adaptive feature selection and regime-specific modeling rather than relying on traditional technical indicators, as well as indicating the critical importance of robust out-of-sample testing in the development of a model.

Keywords: High-Frequency Data; Technical Indicators; Machine Learning; Stock Price Prediction; Risk-Adjusted Performance; Random Forest Regression

*Email: akash.deep@ttu.edu

1 Introduction

Accurate prediction of the stock market remains a fundamental yet highly challenging objective in financial research due to the volatility, noise, and stochasticity in financial markets. As Aldridge [2013] point out, the increasing prevalence of high-frequency trading (HFT), where trades are executed within milliseconds, has intensified the demand for predictive models that can rapidly adapt to market fluctuations and structural complexity. However, developing such models requires overcoming significant hurdles, including the inherent noise in high-frequency data and the rapid shifts in market sentiment, as documented by Gu et al. [2020].

A central theoretical debate in financial economics revolves around the effectiveness of technical indicators in predicting prices. The Efficient Market Hypothesis (EMH) proposed by Fama [1970] suggests that asset prices fully incorporate all available information, rendering historical price-based signals ineffective for forecasting. However, the continued widespread use of technical indicators by traders raises questions about the validity of this assumption, particularly in short-term, high-frequency contexts. As Barberis [2003] point out, technical analysis may capture behavioral biases, such as herding and overconfidence, that contribute to transient market inefficiencies being manifested in the prices. These biases are particularly pronounced at the minute-level, where noise traders (market participants who rely on historical patterns rather than fundamental analysis) may introduce temporary mispricings that machine learning models could exploit.

Machine learning (ML) has emerged as a powerful tool for predicting stock prices, enabling the identification of nonlinear dependencies and complex relationships within historical data. Traditional statistical methods, such as autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) models, often struggle to capture the intricate price dynamics of high-frequency markets, due to their assumptions of linearity. By contrast, ML models, such as random forest regression (RFR), support vector regression (SVR), and gradient boosting, have demonstrated improved predictive performance in financial applications [Derbentsev et al., 2020]. However, their effectiveness is highly contingent on the feature selection, particularly in high-frequency trading environments where the dominance of market noise presents a formidable challenge.

Technical analysis, as outlined by Murphy [1999], employs historical price and volume data through indicators such as Bollinger Bands, exponential moving averages (EMA), and the Commodity Channel Index (CCI), to detect trends and signal potential price reversals. These indicators aim to reflect aggregate market sentiment and trader behavior. However, their effectiveness in HFT is still debated. Studies such as Abrol et al. [2016] suggest that traditional technical indicators often generate unreliable signals in high-frequency environments, where rapid price fluctuations introduce significant noise. Although recent research has examined the integration of technical indicators with machine learning models [Zanc et al., 2019, Fischer and Krauss, 2018], much of this work has focused on daily or hourly data, leaving the complexities of minute-level stock price movements relatively unexplored [Zhang, 2010].

In the present paper, we assess the predictive and risk management performance of random forest regression models augmented with technical indicators for high-frequency stock price prediction. Building on previous research that primarily focuses on daily or hourly data, we extend the analysis to minute-level data, incorporating advanced risk-adjusted performance metrics. This allows us to examine the interplay between technical indicators and the effects of the microstructure of the market, providing new insights into their role in high-frequency trading.

This study tests the hypothesis that while incorporating technical indicators can improve risk-adjusted performance, their effectiveness at prediction diminishes in volatile, high-frequency environments where noise dominates the signal. In particular, we expect that primary price-based

features will contribute more significantly to the predictions of the model than the technical indicators will, aligning with prior evidence for their limited predictive power in short-term trading. In addition, we evaluate the alignment of our findings with the Efficient Market Hypothesis (EMH) by analyzing in-sample versus out-of-sample performance. Our findings provide empirical support for the semi-strong form of the EMH, that while technical indicators may be able to briefly exploit market inefficiencies, their predictive power is limited.

Unlike many of the existing studies, which primarily evaluate technical indicators in daily or hourly trading, our paper is among the first to systematically assess their effectiveness at the minute level, a granularity where the effects of microstructures of the market and noise dominate. Furthermore, prior studies have predominantly relied on conventional evaluation metrics, such as root mean squared error (RMSE) and R-squared (R^2), which provide limited insight into risk-adjusted performance. In contrast, our study employs advanced risk-reward measures, including the Rachev ratio and the gains-loss ratio, offering a more comprehensive evaluation in high-frequency contexts of trading strategies based on machine learning [Cheridito and Kromer, Forthcoming]. By combining insights from technical analysis, machine learning, behavioral finance, and advanced risk management, this paper provides actionable implications for both academics and practitioners seeking to refine predictive modeling techniques for financial markets.

Our findings indicate that while technical indicator-augmented models obtain superior risk-adjusted metrics, when it comes to generating excess returns, they perform worse than a simple buy-and-hold strategy. Following the framework established by Barberis [2003], our results suggest that while technical indicators can enhance risk management, they may not provide sufficient predictive power to consistently outperform baseline strategies in high-frequency environments dominated by market noise and sentiment-driven trading behavior. These insights emphasize the importance of selective feature engineering, regime-aware modeling, and adaptive risk management techniques in the application of machine learning to financial markets, so as to improve the stability of the predictions in high-frequency contexts.

2 Literature Review

Predicting stock prices has been a long-standing challenge due to the volatility and complexity of the market. The Efficient Market Hypothesis (EMH) suggests that prices fully reflect all available information, leaving little room for prediction [Fama, 1970]. However, behavioral finance research has identified systematic deviations from market efficiency, particularly in high-frequency contexts where noise traders may rely heavily on technical indicators [Barberis, 2003]. This tension between rational finance and behavioral finance provides a motivation for evaluating the effectiveness of such technical indicators, as noise traders lacking access to fundamental data may disproportionately rely on technical indicators, potentially creating temporary market inefficiencies [Shleifer and Vishny, 1997].

Advances in machine learning (ML) and the increasing availability of high-frequency trading (HFT) data have made possible the empirical investigation of these theoretical predictions. Traditional econometric models, such as ARIMA and GARCH, were initially used for forecasting stock prices, but often struggled with nonstationary data and volatility clustering, as noted by Zhang [1998] and Patel et al. [2015]. These limitations pointed to the need for integrating ML techniques with traditional financial models in order to improve their predictive accuracy.

In the mid-1990s, ensemble methods, such as random forest, were developed, demonstrating robustness in handling high-dimensional datasets and reducing overfitting through bagging [Ho, 1995]. Recent studies have demonstrated the role of ML in enabling adaptive strategic behaviors

on the part of high-frequency traders. By leveraging tools like genetic algorithms, traders can process complex information about the microstructure of the market and optimize their trading strategies in real time, significantly enhancing their profitability under varying conditions [Arifovic et al., 2022]. The interaction between the speed of the trading and the efficiency of the market has also been explored, finding a hump-shaped relation between speed, efficiency, and the profitability of the trader.

By the early 2000s, studies like Bollinger [2002] began exploring technical indicators, such as Bollinger Bands (BBs), to gauge market trends and overbought or oversold conditions. Meanwhile, the Commodity Channel Index (CCI) and Exponential Moving Average (EMA) emerged as widely used tools for capturing short-term price movements [Lambert, 1983, Murphy, 1999]. However, the standalone use of these indicators often yielded inconsistent results, particularly in noisy and volatile environments, such as HFT [Zhang, 2010].

As ML techniques advanced, studies in the 2010s began integrating technical indicators with ML models to improve the predictive performance. For instance, Fischer and Krauss [2018] demonstrated that combining technical indicators with LSTM networks could reduce noise in high-frequency stock data and enhance the accuracy of the predictions. Gu et al. [2020] expanded on this by showing that ML can uncover market inefficiencies, though these tend to be temporary and limited in nature. Their work emphasized the importance of robust out-of-sample testing and careful feature selection in predictive modeling.

Despite these advances, challenges, such as overfitting and generalization, remained. Researchers like Agrawal et al. [2019] emphasized the importance of domain-specific feature selection to mitigate these problems, while Lim and Zohren [2021] emphasized the need for dynamic models capable of adapting to changing market conditions. Akyildirim et al. [2023] demonstrated that Random Forest models excel at identifying nonlinear patterns in data and perform consistently across different time scales, making them particularly suitable for high-frequency stock price forecasting, where complex relationships exist.

By the early 2020s, the focus shifted toward hybrid strategies combining multiple technical indicators and ML techniques. Zanc et al. [2019] explored the integration of BBs with LSTM networks, showing improvements in predictive accuracy under volatile market conditions. At the same time, studies began addressing the limitations of traditional evaluation metrics, such as the Sharpe and Sortino ratios, which often assume that the returns are normally distributed. Advanced risk-reward metrics, such as the Rachev and modified Rachev ratios, were introduced to provide a more nuanced understanding of how a model performs in volatile environments [Cheridito and Kromer, Forthcoming].

Recent work has increasingly focused on high-frequency data and its unique challenges. Kearns and Nevmyvaka [2013] highlighted the difficulties of extracting meaningful signals from noisy HFT data, while O’Hara [2015] emphasized that market microstructure takes on heightened importance at very fast speeds. These studies underscored the need for models that balance predictive power with robustness against market noise.

Despite substantial progress, significant gaps remain in the literature. Much of the existing work has focused on lower-frequency data, leaving minute-level and tick-level observations underexplored [Zhang, 2010]. Advanced risk-reward metrics, though proposed, have seen limited application in HFT contexts. Hybrid strategies combining multiple technical indicators have shown promise, but their incremental benefits over simpler models are not well-documented. Generalization challenges persist, particularly in HFT settings, where fleeting arbitrage opportunities and high levels of noise increase the risk of overfitting.

This paper contributes to the field by systematically evaluating the predictive and risk-adjusted performance, in an HFT context, of random forest regression models combined with technical

indicators. It incorporates advanced risk–reward metrics to provide a comprehensive assessment of model performance. This paper also addresses generalization issues through rigorous validation techniques and highlights the limited utility of technical indicators in highly volatile settings. By combining technical analysis, ML, and risk management, this paper offers actionable insights for practitioners and researchers aiming to refine predictive modeling in financial markets.

3 Method

In this section, we describe the data acquisition process, the computation of the technical indicators, the ML model (random forest regressor), and the trading simulation framework. The decisions made at each step are guided by the need to rigorously assess the impact of technical indicators on stock price prediction using random forest.

3.1 Data Acquisition and Preprocessing

The dataset used in this study consists of minute-level historical stock data for the SPY (S&P 500 ETF), covering the period from April 2024 to September 2024. The data includes essential fields such as the opening, high, low, and close prices, as well as the trading volume, for each minute. The data were obtained from the Bloomberg Terminal, ensuring high accuracy and reliability [Bloomberg L.P., 2024]. Each data point is timestamped in Central Time (CT), and the dataset covers the typical US stock market hours from 9:30 AM to 4:00 PM Eastern Time (ET), adjusted for daylight savings time.

Additionally, the 10-year US Treasury yield is incorporated as a proxy for the risk-free rate, a crucial factor in calculating excess returns. These data are reported daily and were also sourced from the Bloomberg Terminal, spanning the same time frame as the SPY data [Pastor and Stambaugh, 2003].

3.1.1 Log Returns and Volatility

To normalize the stock price data and reduce the effects of scale, we compute the log returns for the opening, high, low, and close prices. Log returns are preferred in financial time series due to their ability to capture percentage changes and handle volatility over time [Box et al., 2015]. The log return for a price series P_t is

$$\text{log_return}_t = \log\left(\frac{P_t}{P_{t-1}}\right), \quad (1)$$

where P_t is the price at time t . This process is also defined for the opening, high, low, and closing prices, with the resulting log returns stored as additional columns in the dataset. Additionally, we compute rolling Z -scores for the trading volume to capture anomalies in the volume. The rolling Z -score of the volume is [Box et al., 2015]

$$\text{volz}_t = \frac{\text{volume}_t - \text{mean}(\text{volume})}{\text{std}(\text{volume})}, \quad (2)$$

where the mean and standard deviation are computed over a rolling window of 60 minutes.

The 10-year US Treasury yield, provided on a daily basis, is used to compute a per-minute risk-free rate, which is necessary for calculating excess returns. For any minute t within a trading day d , the transformation from the daily yield to a per-minute rate is given by

$$r_{\text{per-minute}}(t) = (1 + r_{\text{daily}}(d))^{\frac{1}{1440}} - 1, \quad (3)$$

where 1440 is the number of minutes in a day and $r_{\text{daily}}(d)$ is the daily risk-free rate derived from the most recently available Treasury yield prior to day d . This ensures that each minute’s risk-free rate reflects the prevailing daily rate for its trading day.

3.1.2 Data Filtering and Splitting

The dataset is filtered to focus on regular market trading hours, between 10:00 AM and 3:30 PM CT, to avoid periods of low liquidity, such as pre-market and after-hours trading [McGroarty et al., 2019]. The filtered dataset is then split into training and testing sets, with 80% of the data allocated to training and 20% to testing. The splitting is time-ordered to preserve the temporal nature of the stock price data and avoid data leakage.

The processed dataset is used for computing a set of technical indicators, which serve as input features for the ML models described in subsequent sections. The computed technical indicators include the simple moving average (SMA), EMA, moving average convergence divergence (MACD), Relative Strength Index (RSI), and others, as detailed below.

3.2 System Architecture

Figure 1 presents a comprehensive view of our ML-based trading system, illustrating the interconnections between the processing of the data, the development of the model, and the evaluation of the performance.

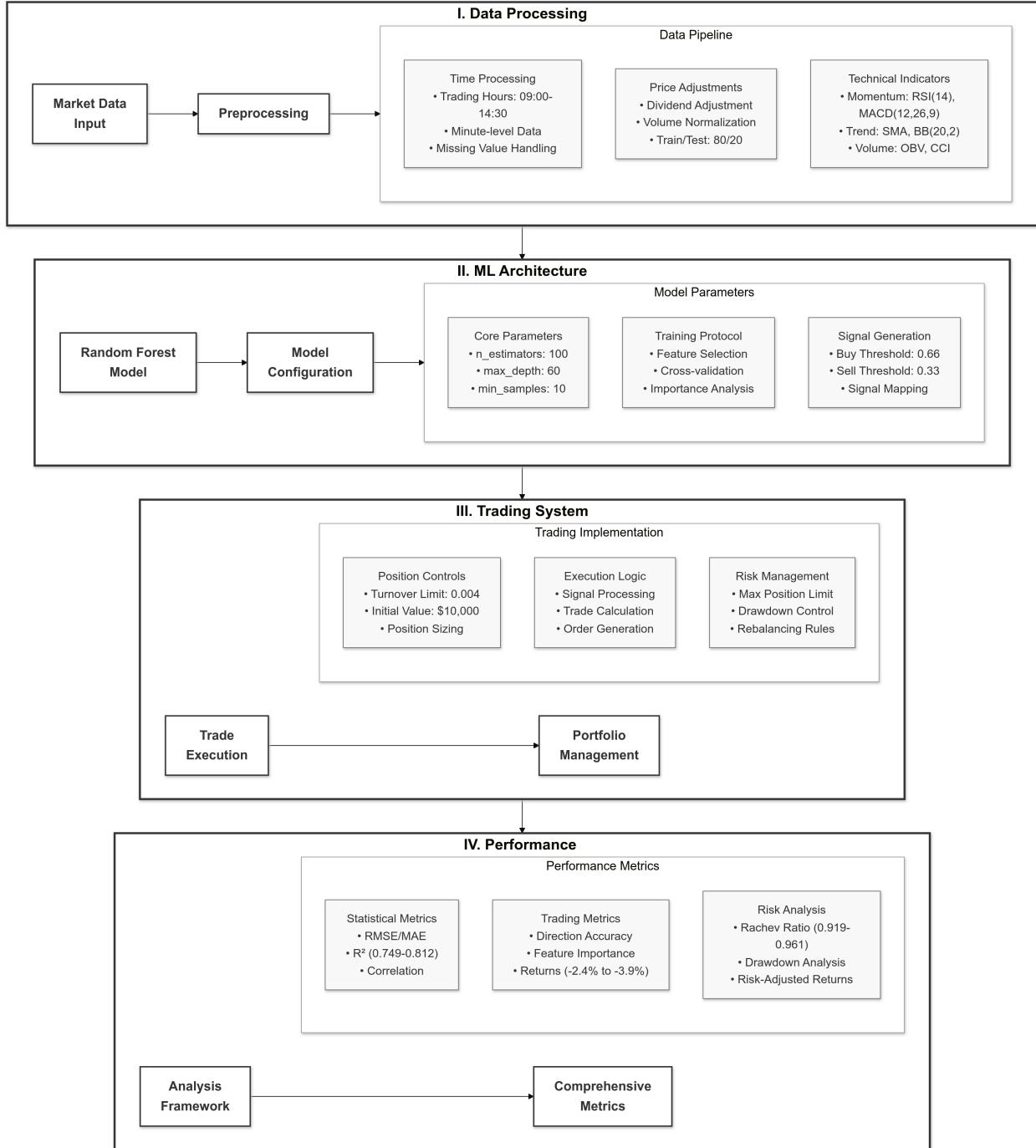


Figure 1: Architecture of the ML-based trading system. There are four integrated phases: (I) Data Processing: handling minute-level SPY data (09:00–14:30), incorporating dividend adjustments, and computing technical indicators including RSI(14), MACD(12,26,9), and Bollinger Bands(20,2); (II) ML Architecture: Random Forest implementation with the specific hyperparameters ($n_estimators=100$, $\text{max_depth}=60$) and quantile-based signal generation (buy: 0.66, sell: 0.33); (III) Trading System: real-time position management with turnover constraints (0.004) and initial capital allocation (\$10,000); and (IV) Performance Analysis: comprehensive evaluation using statistical metrics (the range of values of R^2 is 0.749–0.812) and risk-adjusted measures (Rachev ratio: 0.919–0.961). This system is an example of a practical integration of traditional technical analysis with modern ML approaches, while emphasizing risk management and computational efficiency in high-frequency trading contexts.

Our implementation follows a systematic approach where the data preprocessing feeds into the ML pipeline, which in turn feeds into the trading decisions. This framework incorporates comprehensive risk management and performance evaluation, ensuring robust validation of the strategy's effectiveness. Each component is optimized for high-frequency trading, with particular attention to computational efficiency and real-time processing.

3.3 Technical Indicators

To capture diverse aspects of market behavior, we selected a set of widely recognized technical indicators, each chosen for its unique contribution to predicting price movements or managing risk. These technical indicators encompass a variety of trend-following, momentum, and volume-based metrics, enabling a robust, multi-faceted analysis of minute-level price movements.

For instance, the EMA and MACD offer insights into the strength and direction of a trend, whereas BBs and the RSI gauge the volatility and overbought/oversold conditions, respectively [Murphy, 1999]. The average directional index (ADX) measures the robustness of a trend, the on-balance volume (OBV) measures the volume flow, and the CCI detects cyclical price movements [Lambert, 1983]. By combining these technical indicators, we aimed to create a feature set capable of reflecting both short-term and long-term market dynamics, thus enhancing the predictive accuracy and enabling nuanced risk management [Zanc et al., 2019].

Unless otherwise specified, all non-trivial technical indicator formulae presented are derived from the seminal work [Murphy, 1999].

3.3.1 Simple Moving Average (SMA)

The SMA smooths price data by averaging the closing prices over a window of N periods:

$$\text{SMA}_{N,t} = \frac{1}{N} \sum_{i=0}^{N-1} C_{t-i}, \quad (4)$$

where C_t denotes the closing price at time t . In our implementation, the current price is normalized by the SMA:

$$\hat{\text{SMA}}_{N,t} = \frac{C_t}{\text{SMA}_{N,t}}. \quad (5)$$

This ensures scale invariance and helps the model better learn from the price data.

3.3.2 Exponential Moving Average (EMA)

The EMA places more weight on recent prices, making it more responsive to changes in prices. It is calculated recursively as follows:

$$\text{EMA}_t = \alpha C_t + (1 - \alpha) \text{EMA}_{t-1}, \quad (6)$$

where $\alpha = \frac{2}{N+1}$ is the smoothing factor for a window size N . In our implementation, the EMA is normalized similarly to the SMA:

$$\hat{\text{EMA}}_t = \frac{C_t}{\text{EMA}_t}. \quad (7)$$

This ratio stabilizes the feature and makes it more useful for prediction.

3.3.3 Moving Average Convergence Divergence (MACD)

The MACD measures the difference between short-term and long-term EMAs. It is computed as follows:

$$\text{MACD}_t = \text{EMA}_{12,t} - \text{EMA}_{26,t}. \quad (8)$$

The signal line SIG_t is a 9-period EMA of the MACD line. We use the following ratio to normalize the MACD:

$$r_{\text{MACD},t} = \frac{\text{MACD}_t - \text{SIG}_t}{0.5(|\text{MACD}_t| + |\text{SIG}_t|)}. \quad (9)$$

This ensures that large fluctuations in the MACD do not overwhelm the model.

3.3.4 Relative Strength Index (RSI)

The RSI is a momentum oscillator that measures the speed and change of price movements [Wilder, 1978]. It is computed as follows:

$$\text{RSI}_t = 100 - \frac{100}{1 + \frac{\text{avg_gain}_t}{\text{avg_loss}_t}}, \quad (10)$$

where avg_gain_t and avg_loss_t are the exponentially smoothed averages of the gains and losses over a window of 14 periods. The RSI ranges from 0 to 100, identifying possible overbought and oversold conditions.

3.3.5 Bollinger Bands (BBs)

Bollinger bands are volatility bands placed two standard deviations above and below a moving average. They are defined by

$$\text{UBB}_t = \text{SMA}_{N,t} + 2\sigma_t, \quad \text{LBB}_t = \text{SMA}_{N,t} - 2\sigma_t, \quad (11)$$

where σ_t is the standard deviation of the prices over the last N periods [Bollinger, 2002]. The normalized BB percentage is

$$\text{BB\%}_t = \frac{C_t - \text{LBB}_t}{\text{UBB}_t - \text{LBB}_t}. \quad (12)$$

This captures where the price sits within the volatility bands.

3.3.6 Stochastic Oscillator (SO)

The stochastic oscillator (SO) measures the relative position of the closing price compared to the high–low range over a specified period (typically 14 periods). It is computed as follows:

$$\%K_t = 100 \times \frac{C_t - L_{14,t}}{H_{14,t} - L_{14,t}}, \quad (13)$$

where $L_{14,t}$ and $H_{14,t}$ denote the lowest and highest prices over the last 14 periods. The slow stochastic oscillator $\%D_t$ is a 3-period moving average of $\%K_t$.

3.3.7 Fibonacci Retracement (Fib)

The Fibonacci retracement is used to identify potential support and resistance levels in a price trend. For a window N , the retracement level is

$$R_t = \frac{H_{N,t} - C_t}{H_{N,t} - L_{N,t}}, \quad (14)$$

where $H_{N,t}$ and $L_{N,t}$ are the highest and lowest prices over the window. We use common Fibonacci levels (0.236, 0.382, 0.500, 0.618, 0.764) to identify potential reversal points.

3.3.8 Average Directional Index (ADX)

The ADX measures the strength of a trend, regardless of its direction. The ADX is derived from the directional movement indicators DI_t^+ and DI_t^- :

$$\text{ADX}_t = \frac{|DI_t^+ - DI_t^-|}{DI_t^+ + DI_t^-}. \quad (15)$$

The directional movement indicators DI_t^+ and DI_t^- are normalized by the average true range (ATR).

3.3.9 On-Balance Volume (OBV)

The OBV is a cumulative indicator that sums the volumes, depending on whether the price is rising or falling:

$$\text{OBV}_t = \text{OBV}_{t-1} + \text{sgn}(C_t - C_{t-1})V_t, \quad (16)$$

where V_t is the trading volume at time t , and the signum function determines the direction of the volume flow.

3.3.10 Windowed Relative OBV (WROBV)

The windowed relative OBV (WROBV) is a modified version of the OBV. It is the weighted sum of the values, for a rolling window of size N , of the cumulative OBV. This smooths out the indicator:

$$\text{WROBV}_t = \frac{\sum_{i=0}^{N-1} \text{OBV}_{t-i}}{\sum_{i=0}^{N-1} V_{t-i}}. \quad (17)$$

This rolling normalization prevents the OBV from becoming excessively large, and focuses on recent price–volume dynamics.

3.3.11 Commodity Channel Index (CCI)

The CCI measures the deviation of the typical price from its moving average:

$$p_t = \frac{H_t + L_t + C_t}{3}. \quad (18)$$

The CCI is given by

$$\text{CCI}_t = \frac{p_t - \text{SMA}_{N,t}}{0.015 \times \text{MAD}_t}, \quad (19)$$

where MAD_t is the mean, over a rolling window of size N , of the absolute deviations of p_t .

3.3.12 Ichimoku Cloud (Ichimoku)

The Ichimoku Cloud is a comprehensive technical indicator that provides a holistic view of support, resistance, the direction of the trend, and momentum [Patel, 2010]. It has five main components:

- Tenkan-sen (Conversion Line): This line is a short-term indicator calculated as the midpoint of the highest high and the lowest low over the past N periods:

$$\text{Tenkan}_t = \frac{\max(H_{t-N}, \dots, H_t) + \min(L_{t-N}, \dots, L_t)}{2}, \quad (20)$$

where H_t and L_t denote the high and low prices at time t , respectively. Typically, $N = 9$.

- Kijun-sen (Base Line): The base line is a longer-term indicator calculated similarly to the Tenkan-sen but over a longer window M :

$$\text{Kijun}_t = \frac{\max(H_{t-M}, \dots, H_t) + \min(L_{t-M}, \dots, L_t)}{2}. \quad (21)$$

This line provides a measure of medium-term momentum, with $M = 26$ being a common value.

- Senkou Span A (Leading Span A): Senkou Span A is the midpoint between the Tenkan-sen and Kijun-sen, plotted M periods ahead:

$$\text{Senkou A}_t = \frac{\text{Tenkan}_t + \text{Kijun}_t}{2} \quad (\text{shifted forward by } M \text{ periods}). \quad (22)$$

This span, along with the following Senkou Span B, forms the Ichimoku Cloud.

- Senkou Span B (Leading Span B): This span is the midpoint of the highest high and lowest low over the past L periods and is also plotted M periods ahead:

$$\text{Senkou B}_t = \frac{\max(H_{t-L}, \dots, H_t) + \min(L_{t-L}, \dots, L_t)}{2} \quad (\text{shifted forward by } M \text{ periods}). \quad (23)$$

The area between Senkou Span A and Senkou Span B is shaded to form the ‘cloud,’ which can act as dynamic support or resistance.

- Chikou Span (Lagging Span): The Chikou Span is the current closing price plotted M periods in the past:

$$\text{Chikou}_t = C_t \quad (\text{shifted backward by } M \text{ periods}). \quad (24)$$

This line provides a lagging indication of price action and helps confirm the direction of a trend.

The Ichimoku Cloud provides a visual representation of support and resistance, the direction of a trend, and momentum. The interaction between the price and the cloud helps identify potential reversals or continuations in the trend. In our implementation, we calculate all five components of the Ichimoku cloud and incorporate the leading spans (Senkou A and Senkou B) as features in the machine learning model.

3.4 Random Forest and Validation

The underlying predictive model in our framework uses a random forest regressor (RFR), which is an ensemble learning method that aggregates predictions from multiple decision trees to capture complex, non-linear relations in high-frequency financial data [Ho, 1995, Breiman, 2001, Buitinck et al., unpublished]. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote our training dataset, where $\mathbf{x}_i \in \mathbb{R}^p$ denotes the feature vector consisting of technical indicators and price-based features at time i , and $y_i \in \mathbb{R}$ denotes the corresponding log return.

The RFR constructs an ensemble of B decision trees, where each tree T_b is trained on a bootstrap sample \mathcal{D}_b drawn with replacement from \mathcal{D} . For a given input vector \mathbf{x} , the model's prediction is

$$\hat{f}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B T_b(\mathbf{x}), \quad (25)$$

where $T_b(\mathbf{x})$ denotes the prediction of the b -th tree. Each individual tree is constructed by recursively partitioning the feature space to minimize the mean squared error (MSE):

$$\text{MSE}(t) = \frac{1}{|\mathcal{D}_t|} \sum_{i \in \mathcal{D}_t} (y_i - \bar{y}_t)^2, \quad (26)$$

where \mathcal{D}_t denotes the set of training samples at node t , and \bar{y}_t is the mean response value in node t .

Our implementation employs scikit-learn's RandomForestRegressor with the following parametrization:

$$\Theta = \{\theta_B, \theta_d, \theta_s, \theta_f, \theta_l, \theta_r\}, \quad (27)$$

where $\theta_B = 100$ (n_estimators), $\theta_d = 60$ (max_depth), $\theta_s = 10$ (min_samples_split), $\theta_f = \text{'log2'}$ (max_features), $\theta_l = 1$ (min_samples_leaf), and $\theta_r = 42$ (random_state). This configuration performs cross-validation through Out-of-Bag (OOB) sampling [Hastie et al., 2009], where approximately one-third of the observations are automatically held out during the training of each tree, serving as a built-in validation set.

To determine the signal, we employ a quantile-based thresholding mechanism. Let $\hat{f}(\mathbf{x}_t)$ be the model's prediction at time t . During training, we compute threshold values $q_{0.33}$ and $q_{0.66}$, which are the 33rd and 66th percentiles of the model's predictions on the training set. The signal function $s : \mathbb{R} \rightarrow \{\text{"sell"}, \text{"hold"}, \text{"buy"}\}$ is defined by

$$s(\hat{f}(\mathbf{x}_t)) = \begin{cases} \text{"buy"} & \text{if } \hat{f}(\mathbf{x}_t) \geq q_{0.66} \\ \text{"hold"} & \text{if } q_{0.33} < \hat{f}(\mathbf{x}_t) < q_{0.66} \\ \text{"sell"} & \text{if } \hat{f}(\mathbf{x}_t) \leq q_{0.33} \end{cases} \quad (28)$$

The importance of a feature is computed using the mean decrease in impurity across all trees:

$$I_j = \frac{1}{B} \sum_{b=1}^B \sum_{t \in \mathcal{T}_b} \Delta \text{MSE}_{t,j} \mathbb{1}(v(t) = j), \quad (29)$$

where \mathcal{T}_b is the set of nodes in tree b , $v(t)$ is the feature used for splitting at node t , and $\Delta \text{MSE}_{t,j}$ is the decrease in MSE achieved by splitting on feature j at node t .

For temporal validation, we employ a chronological partitioning:

$$\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\lfloor 0.8n \rfloor}, \quad \mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=\lfloor 0.8n \rfloor+1}^n, \quad (30)$$

ensuring strict temporal ordering and preventing look-ahead bias. This 80–20 split, combined with the OOB error estimation, provides a robust validation framework that appropriately handles both the ensemble nature of Random Forests and the sequential characteristics of high-frequency trading data. As demonstrated by Hastie et al. [2009], the OOB error estimate is nearly equivalent to leave-one-out cross-validation, providing an unbiased estimate of the test error and making additional k -fold cross-validation unnecessary.

While time-series cross-validation (TSCV), such as rolling or walk-forward validation, is a common approach in the forecasting of financial time-series, its application to high-frequency trading (HFT) remains computationally intensive. Given the ensemble nature of Random Forests and the strict temporal partitioning employed in our study, we rely on Out-of-Bag (OOB) error estimation as an efficient alternative. This method maintains the chronological integrity of the data while avoiding excessive computational overhead. Future research should explore the trade-off between computational feasibility and the robustness benefits of TSCV, particularly in adaptive trading models where market regimes shift dynamically.

3.5 Trading Simulation

We simulate a trading strategy based on the buy, sell, and hold signals generated by the random forest model. The trading simulation starts with an initial value of \$10,000. The following actions are taken based on the predictions of the model:

- **Buy Signal:** If the model predicts an upward price movement, a portion of the available cash is used to buy shares.
- **Sell Signal:** If a downward price movement is predicted, a portion of the holdings is sold.
- **Hold Signal:** If no significant price movement is predicted, no action is taken.

To approximate real-world trading constraints, we impose a turnover constraint limiting position changes in any minute to 0.4% of the portfolio value:

$$\frac{\text{value_traded}}{\text{portfolio_value}} \leq 0.004. \quad (31)$$

This constraint was chosen to align with SPY’s typical daily turnover rate of approximately 3%. By limiting per-minute turnover to 0.4%, our simulation ensures that total daily changes in the position remain within realistic bounds, given the ETF’s observed liquidity characteristics. All trades are executed at minute-end closing prices. While this implementation provides realistic control of the sizes of the positions, it is somewhat optimistic, as it does not take into account bid–ask spreads, commission costs, or potential price impact. These limitations should be considered when interpreting the performance of the strategy.

3.6 Evaluation Metrics

We assess the performance of the random forest model using a comprehensive set of metrics that evaluate both the accuracy of the predictions and the risk-adjusted returns:

- **Root Mean Squared Error (RMSE):** this is the square root of the average of the squares of the differences between the predicted and actual returns:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}. \quad (32)$$

A lower RMSE indicates a better accuracy.

- **Mean Absolute Error (MAE):** This is the average of the absolute differences between the predicted and actual returns, offering an intuitive measure of the model's accuracy.
- **R-squared (R^2):** This is the proportion of the variance in the target variable explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (33)$$

where \bar{y} is the mean of the actual returns. A higher R^2 indicates a better performance of the model.

- **Trend Accuracy:** This evaluates the model's ability to predict the direction (up or down) of price movements:

$$\text{Trend Accuracy} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(\text{sign}(y_i) = \text{sign}(\hat{y}_i)), \quad (34)$$

where $\mathbf{1}$ is an indicator function returning 1 if the predicted direction matches the actual direction, and 0 otherwise.

- **Sharpe Ratio:** This assesses the risk-adjusted performance of the trading strategy:

$$\text{Sharpe Ratio} = \frac{\mathbb{E}[r_p - r_f]}{\sigma_p}, \quad (35)$$

where r_p is the asset return, r_f is the risk-free rate, and σ_p is the standard deviation of the returns. A higher Sharpe ratio indicates better risk-adjusted returns.

- **Maximum Drawdown:** This is the largest peak-to-trough decline in the value of the asset over the testing period:

$$\text{Max Drawdown} = \max_{t \in T} \left(\frac{\text{peak}_t - \text{trough}_t}{\text{peak}_t} \right). \quad (36)$$

This metric is crucial for evaluating the worst-case performance of the trading strategy during periods of market stress.

- **Sortino Ratio:** This is a variant of the Sharpe ratio that focuses only on downside risk:

$$\text{Sortino Ratio} = \frac{\mathbb{E}[r_p - r_f]}{\text{Downside Deviation}}, \quad (37)$$

where the downside deviation is calculated using only negative returns. This ratio penalizes excessive downside risk more than overall volatility.

These metrics provide a well-rounded evaluation of the RFR's predictive accuracy and its ability to manage risk in the context of an HFT strategy.

3.7 Selection of Risk–Reward Ratios

In selecting risk–reward ratios for this study, we follow the theoretical framework laid out by Cheridito and Kromer [Forthcoming], focusing on ratios that satisfy the following four critical properties:

- **Monotonicity (M):** This property ensures that the reward–risk ratio (RRR) increases as returns increase, for a fixed level of risk. Essentially, this criterion reflects the intuitive idea that ‘more is better.’ Formally, for two random variables X and Y , where $X \geq Y$, we should have $\rho(X) \geq \rho(Y)$.
- **Quasi-Concavity (Q):** Quasi-concavity encourages diversification, ensuring that the ratio prefers averages over extremes. If a reward–risk ratio satisfies this property, this means that a diversified portfolio will generally be preferred over a concentrated risk. Formally, for random variables X and Y , and for any $\lambda \in [0, 1]$, we should have $\rho(\lambda X + (1-\lambda)Y) \geq \min(\rho(X), \rho(Y))$.
- **Scale Invariance (S):** Scale invariance means that the ratio remains unchanged when both the return and the risk of a portfolio are scaled by the same factor. This ensures that the ratio is consistent across different investment sizes; it requires that $\rho(\lambda X) = \rho(X)$ for all positive scalars λ .
- **Distribution-based (D):** This property ensures that the ratio depends only on the distribution of the returns X and not on any specific realization of X . This is essential for generalizing the performance metric across different scenarios and portfolio strategies.

These properties form a robust basis for evaluating performance metrics, ensuring that they promote diversification and reward consistency. Many risk–reward ratios used in the financial literature—such as the Sharpe ratio, Sortino ratio, Rachev ratio, and others—naturally satisfy these criteria. The ratios chosen for this study align with these principles, allowing a comprehensive evaluation of the performance of a portfolio.

Table 1: Risk–reward ratios used in the study.

Ratio	Formula	Description
Sharpe ratio	$\frac{\mathbb{E}[R_p - R_f]}{\sigma_p}$	R_p : Portfolio return, R_f : Risk-free rate, σ_p : Standard deviation of excess returns. Measures the excess return per unit of risk (volatility), highlighting risk-adjusted performance.
Sortino ratio	$\frac{\mathbb{E}[R_p - R_f]}{\sigma_d}$	σ_d : Standard deviation of negative returns (downside risk). Improves on the Sharpe ratio by focusing only on downside risk, penalizing large losses more than fluctuations from gains.
Rachev ratio	$\frac{\mathbb{E}[R_p R_p \geq \text{VaR}_{1-\gamma}]}{\mathbb{E}[R_p R_p \leq \text{VaR}_\beta]}$	VaR : Value-at-Risk, γ : Upper quantile, β : Lower quantile. Measures tail risk by comparing the potential gains in the best-case scenario with the worst-case losses.
Modified Rachev ratio	$\frac{\mathbb{E}[R_p R_p \geq \text{VaR}_{1-\delta}]/\epsilon}{\mathbb{E}[R_p R_p \leq \text{VaR}_\delta]/\gamma}$	δ, ϵ : Additional parameters to refine the evaluation of risk. Extends the Rachev ratio to offer a more granular comparison between upper and lower tails at multiple confidence levels.
Distortion RRR	$\frac{\mathbb{E}[R_p R_p \geq \text{VaR}_{1-\beta}]}{\mathbb{E}[R_p R_p \leq \text{VaR}_\beta]}$	VaR : Value-at-Risk, β : Confidence level. Uses a distortion function to adjust the weights of the gains and losses, allowing flexible risk assessments depending on the investor's preferences.
Gains–Loss ratio	$\frac{\mathbb{E}[R_p R_p > 0]}{\mathbb{E}[R_p R_p < 0]}$	The ratio of the average positive returns over the average negative returns, providing a simple risk–reward comparison.
STAR ratio	$\frac{\mathbb{E}[R_p - R_f]}{\mathbb{E}[R_p R_p \leq \text{VaR}_\alpha]}$	VaR : Value-at-Risk, α : Confidence level. Focuses on tail risk, using the Conditional Value-at-Risk (CVaR), also known as the expected shortfall, to take into account extreme losses.
MiniMax ratio	$\frac{\mathbb{E}[R_p]}{\text{Max Drawdown}}$	Max Drawdown: Largest peak-to-trough decline in portfolio value. Compares the average return to the largest drawdown, focusing on how the strategy performs relative to its worst loss.
Gini ratio	$\frac{\sum_{i=1}^N (2i - N - 1) R_i}{N \sum_{i=1}^N R_i}$	R_i : Sorted returns, N : Number of observations. Measures the inequality in the distribution of returns, analogous to the Gini coefficient used in economics.

4 Results

This section summarizes the performance of the random forest regression (RFR) models with and without technical indicators, compares them to a buy-and-hold benchmark, and discusses their statistical significance. The results include the predictive accuracy, the trading outcomes, the risk-adjusted performance, the contributions made by each feature, and residual analyses.

4.1 Predictive Performance

4.1.1 Training vs. Testing Metrics

We trained and tested 13 RFR models, differing in their inclusion of technical indicators, and compared their performance to a buy-and-hold benchmark. Table 2 presents the root mean square error (RMSE), mean absolute error (MAE), and R^2 for both training and testing sets. Although the models generally achieved strong results in-sample (training R^2 from 0.749 to 0.812), out-of-sample performance deteriorated (testing R^2 in the range -0.020 to -0.016). This discrepancy points to overfitting, consistent with the challenges often encountered when applying ML to minute-level data.

Table 2: Model performance metrics for training and testing.

Model	RMSE		MAE		R^2	
	Train	Test	Train	Test	Train	Test
RFR (no indicators)	0.00021	0.00036	0.00015	0.00024	0.786	-0.020
rfr_boll	0.00021	0.00036	0.00015	0.00024	0.812	-0.016
rfr_ema	0.00022	0.00036	0.00016	0.00024	0.749	-0.019
rfr_rsi	0.00021	0.00036	0.00015	0.00024	0.802	-0.017

All models have comparable RMSEs (0.00036) and MAEs (0.00024) in the test set, indicating little variation in forecasting error. The negative out-of-sample R^2 values for each model confirm that high in-sample fits did not translate into predictive power on unseen data.

4.2 Outcomes of the Trading Strategies

4.2.1 Portfolio Value and Returns

We simulated a trading strategy for each model from August 28, 2024, to October 4, 2024, starting with \$10,000. Figure 2 shows the trajectories of the portfolios, and Table 3 shows the final portfolio values, returns, and major performance ratios. The buy-and-hold strategy ended at \$10,229, which counts as a 0% deviation from the baseline, since it is the baseline in our setting. All RFR-based strategies underperformed.

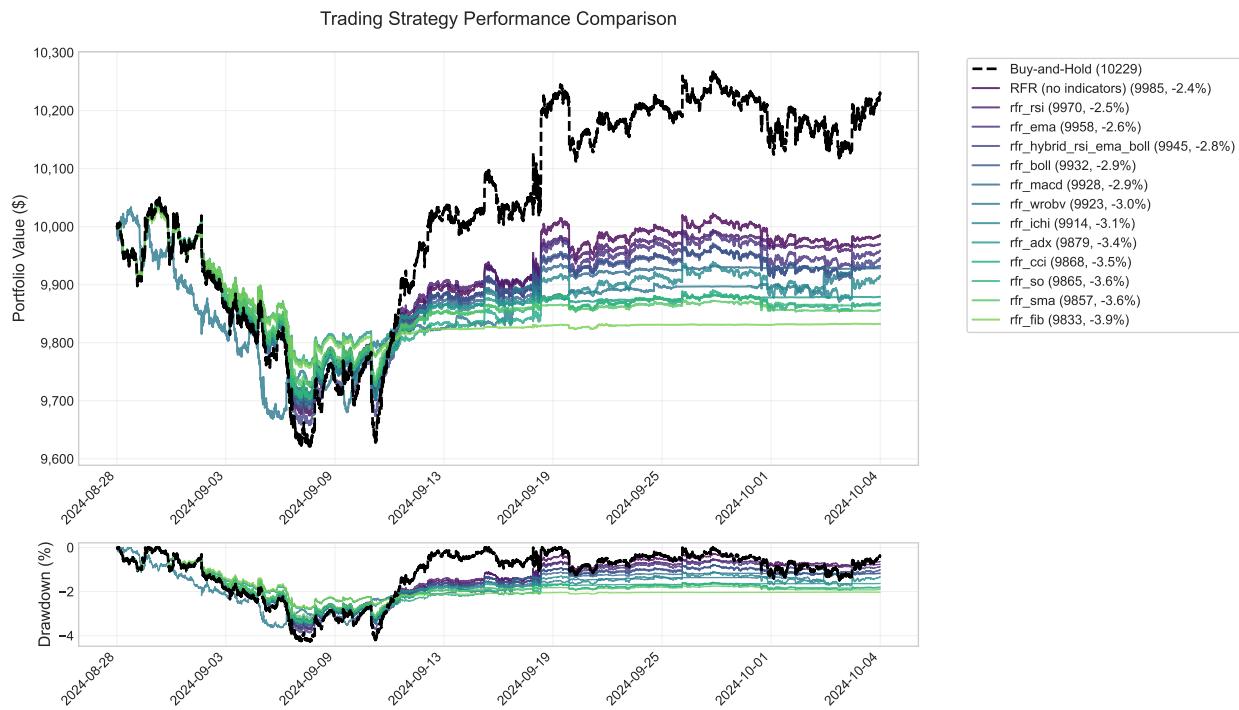


Figure 2: Trajectories of the values of the portfolios for different trading strategies. The buy-and-hold approach ended at \$10,229, while the algorithmic models underperformed to different degrees. Maximum drawdown was around 4%.

Table 3: Summary of trading performance.

Model	Final Value (\$)	Return (%)	Sharpe	Sortino	Rachev
Buy-and-hold	10,229	0.00	—	—	—
RFR (no indicators)	9,985	-2.40	0.0046	0.0047	0.946
rfr_rsi	9,970	-2.50	-0.0015	-0.0018	0.961
rfr_ema	9,958	-2.60	-0.0020	-0.0024	0.961
rfr_hybrid_rsi_ema_boll	9,945	-2.80	-0.0024	-0.0029	0.956
rfr_boll	9,932	-2.90	-0.0033	-0.0040	0.957
rfr_macd	9,928	-2.90	-0.0035	-0.0041	0.953
rfr_wrobd	9,923	-3.00	-0.0041	-0.0046	0.938
rfr_ichi	9,914	-3.10	-0.0040	-0.0048	0.950
rfr_adx	9,879	-3.40	-0.0078	-0.0089	0.937
rfr_cci	9,868	-3.50	-0.0069	-0.0082	0.943
rfr_so	9,865	-3.60	-0.0073	-0.0083	0.939
rfr_sma	9,857	-3.60	-0.0082	-0.0093	0.937
rfr_fib	9,833	-3.90	-0.0116	-0.0133	0.919

Although each model ended below \$10,000, a few (notably, RFR with no indicators or rfr_rsi) performed slightly better than the others in risk-adjusted terms, with Sharpe ratios near 0.00 to 0.0046. None, however, surpassed the buy-and-hold benchmark in absolute returns.

That RFR-based strategies behave worse than the buy-and-hold benchmark can be attributed to transaction costs and market noise, which diminish the effectiveness of short-term trading strategies. While technical indicators provide some value in capturing short-term inefficiencies, the minute-level predictive horizon may not be sufficient to extract profitable trading signals. Additionally, high turnover rates in algorithmic strategies increase trading costs, further eroding potential returns in real-world implementations.

Our results align with prior studies, such as Peng et al. [2021], which found that technical indicators provide limited predictive value in deep learning models trained on daily-level data. Unlike our Random Forest approach, studies leveraging LSTMs and attention-based models have demonstrated better sequence-learning capabilities. However, our findings suggest that even with alternative architectures, the predictive power for high-frequency trading intervals remains constrained due to the market’s microstructural noise.

4.3 Risk-Adjusted Performance

We assessed each strategy using risk metrics such as the Sharpe, Sortino, and Rachev ratios. Figure 3 presents a radar chart comparing the top five models; Figure 4 presents a heatmap of their risk-reward profiles.

Despite the differing results of the different strategies, none of the models yielded Sharpe ratios above 0.0046, a figure significantly below industry standards for viable trading strategies. This suggests that technical indicators alone may not be sufficient for high-frequency trading, as the models struggle to achieve risk-adjusted returns that justify frequent trading. Furthermore, the consistently negative Sortino ratios highlight that these models do not effectively protect against downside risk, reinforcing the argument that ML-based strategies in high-frequency trading environments face structural challenges.

Among the tested models, RFR.RSI and RFR.ICHIMOKU obtained slightly better Rachev ratios (0.919–0.961), suggesting that momentum-based indicators may offer a small advantage in

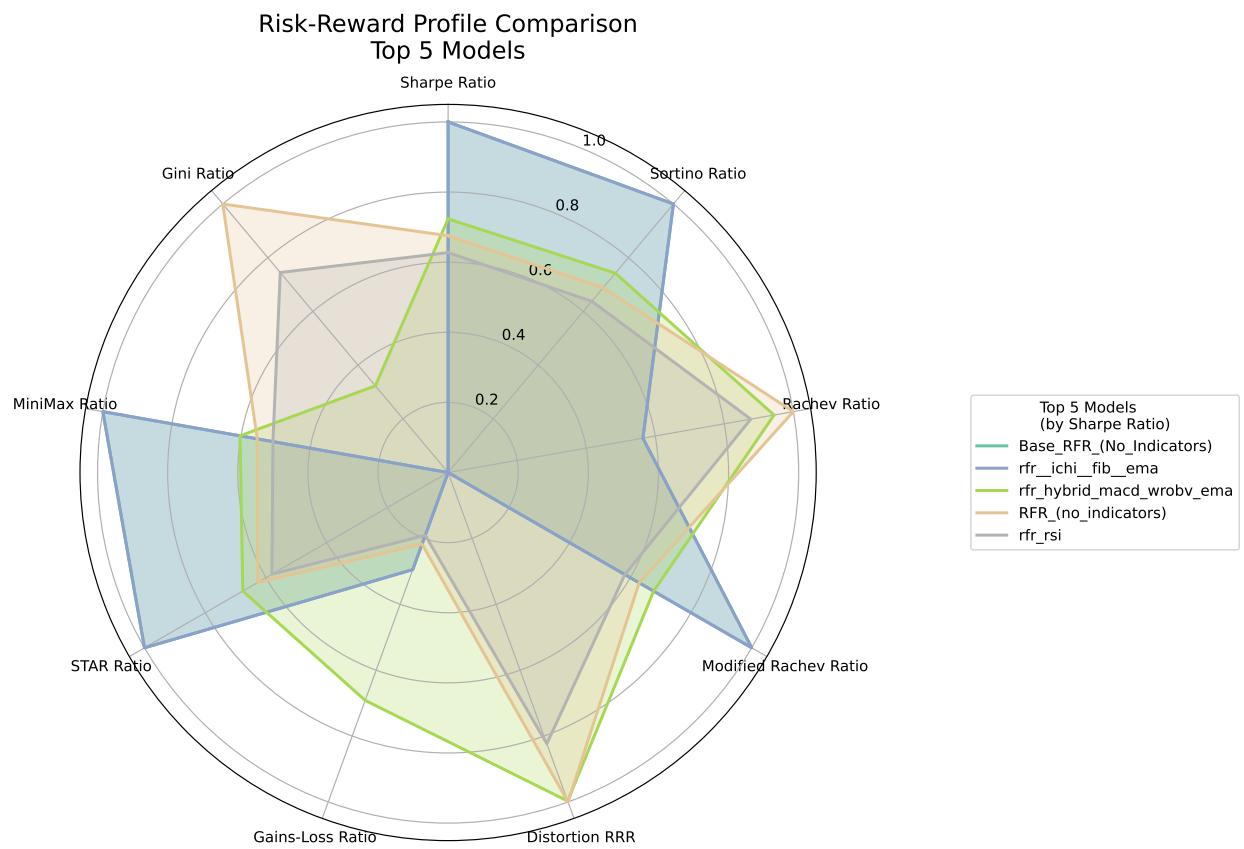


Figure 3: Risk–reward profiles for the top five models, presenting the Sharpe, Sortino, and Rachev ratios.

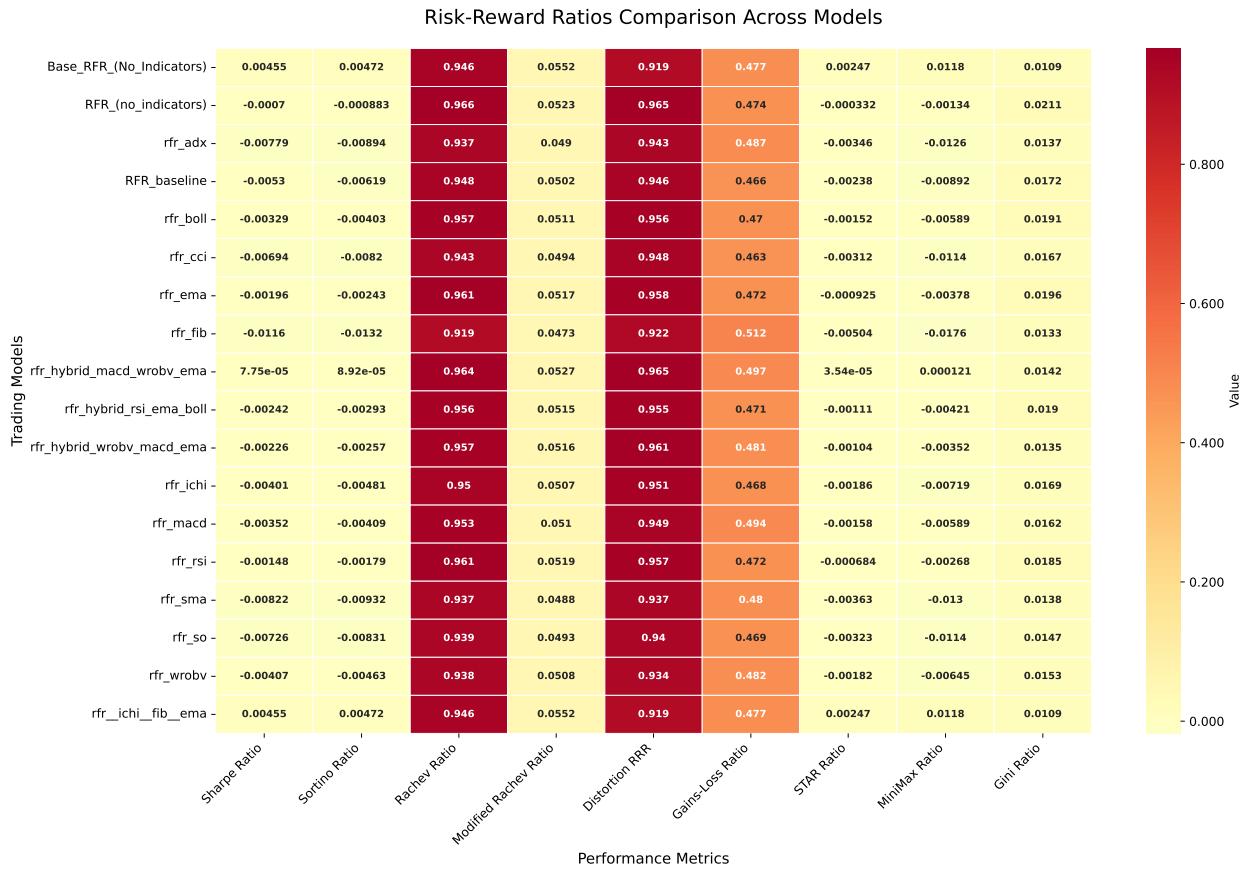


Figure 4: Heatmap comparing Sharpe, Sortino, Rachev, and modified Rachev ratios for all models.

risk-reward trade-offs. However, these improvements were marginal and probably not statistically significant, indicating the need for further research with larger datasets and multi-asset testing.

A deeper analysis of the strategy's performance reveals that most trading losses occurred during periods of heightened market volatility, suggesting that the models struggle to adapt dynamically to shifting volatility regimes. These findings indicate that future research should explore adaptive models that adjust their feature weighting based on changing market conditions. The inability to effectively navigate volatility shocks highlights a key limitation of static ML models in financial applications, reinforcing the need for more flexible approaches that can integrate real-time volatility estimation.

These results challenge the weak form of the Efficient Market Hypothesis (EMH), suggesting that technical indicators may contribute to risk-adjusted decision-making but fail to generate persistent excess returns. This aligns with prior studies that found short-term inefficiencies in financial markets to be highly transient and difficult to exploit systematically. Future work should examine whether alternative data sources, such as order book data, sentiment analysis, or macroeconomic signals, could enhance the predictive power.

4.4 Feature Importance

4.4.1 Base Model

In the base RFR model (Figure 5), the closing, opening, high, low, and volume (normalized as a Z-score) features accounted for over 90% of the total importance. This indicates that raw price and volume data captured most short-term market signals for minute-level trading, consistent with the literature suggesting that in high-frequency contexts, market noise overwhelms many of the usual indicators.

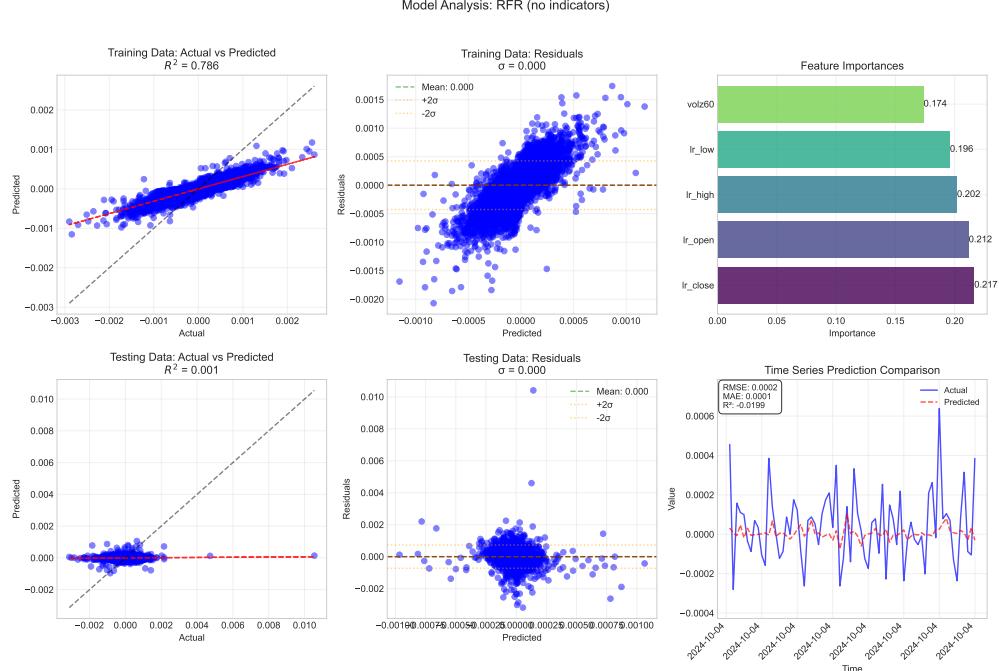


Figure 5: Feature importance distribution for the base RFR model without technical indicators.

4.4.2 Technical Indicators

Adding Bollinger Bands, EMA, or RSI (Figures 6–8) shifted the distribution slightly, with these indicators contributing 14%–18% to the predictive decisions. However, none of these changes substantially improved the out-of-sample accuracy or trading outcomes, implying that traditional indicators do not offer a stable advantage at minute-level frequency.

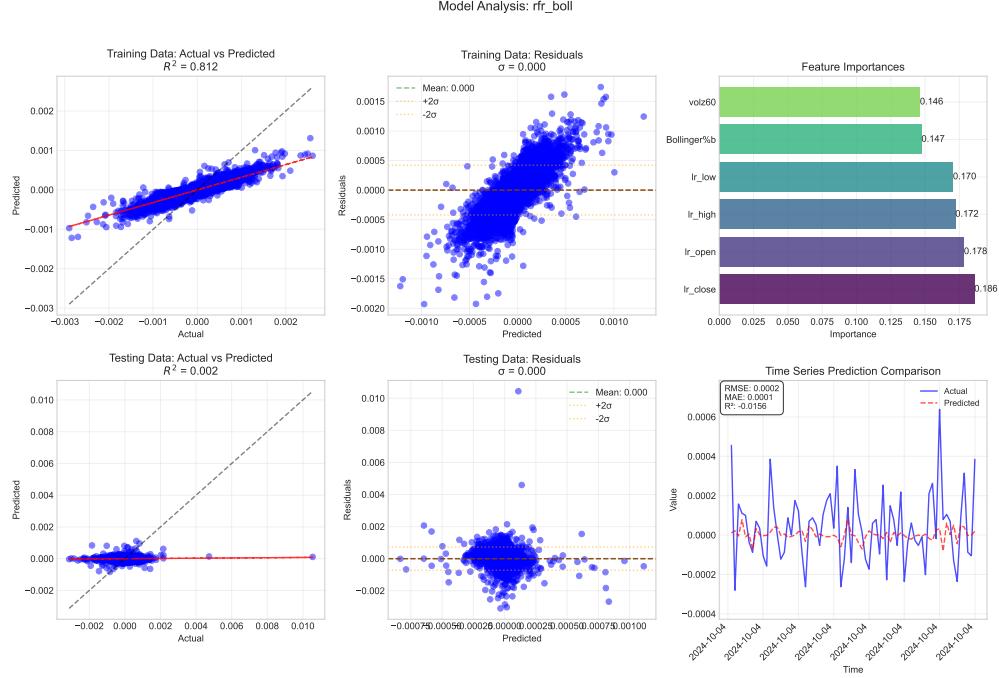


Figure 6: Feature importance distribution for the RFR model including Bollinger Bands.

4.5 Residual Analysis and Directional Accuracy

Residual plots for the base model (Figure 9) showed no strong bias or autocorrelation, suggesting consistent performance within the sample. However, the directional accuracy dropped notably from 80%–87% in training to 48%–50% in testing, again pointing to overfitting. The correlation coefficients also declined from 0.86–0.92 (training) to 0.03–0.06 (testing).

4.6 Comparative Analysis and Statistical Significance

When comparing models that use standard technical indicators (e.g., RSI, EMA, and Bollinger Bands) with those relying only on features based on the raw prices, the former did not exhibit a clear advantage in out-of-sample prediction or final returns. Although hybrid approaches combining multiple indicators slightly reduced the maximum drawdowns, they still failed to outperform simpler RFR models in absolute or risk-adjusted returns.

Statistical tests reinforced these findings: despite high in-sample R^2 , all models obtained negative out-of-sample R^2 . Consistent RMSE and MAE values across variants of the model further suggest that adding technical indicators did not meaningfully reduce forecast errors.

In summary, these results highlight the challenges in exploiting minute-level data with standard technical indicators. While the models fit historical data reasonably well, they struggled to gener-

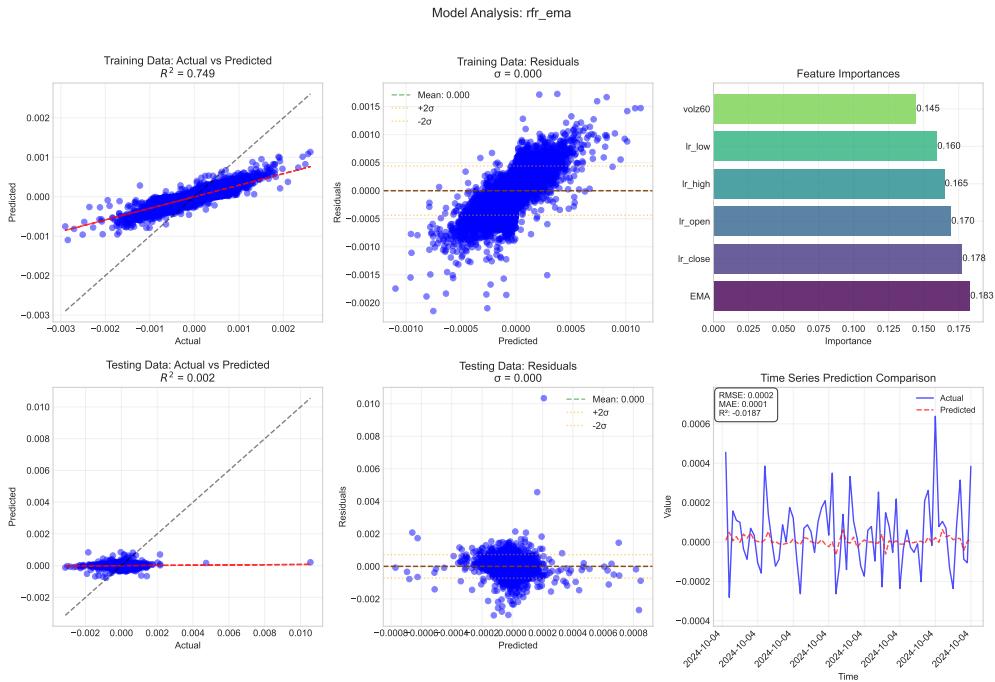


Figure 7: Feature importance distribution for the RFR model including EMA.

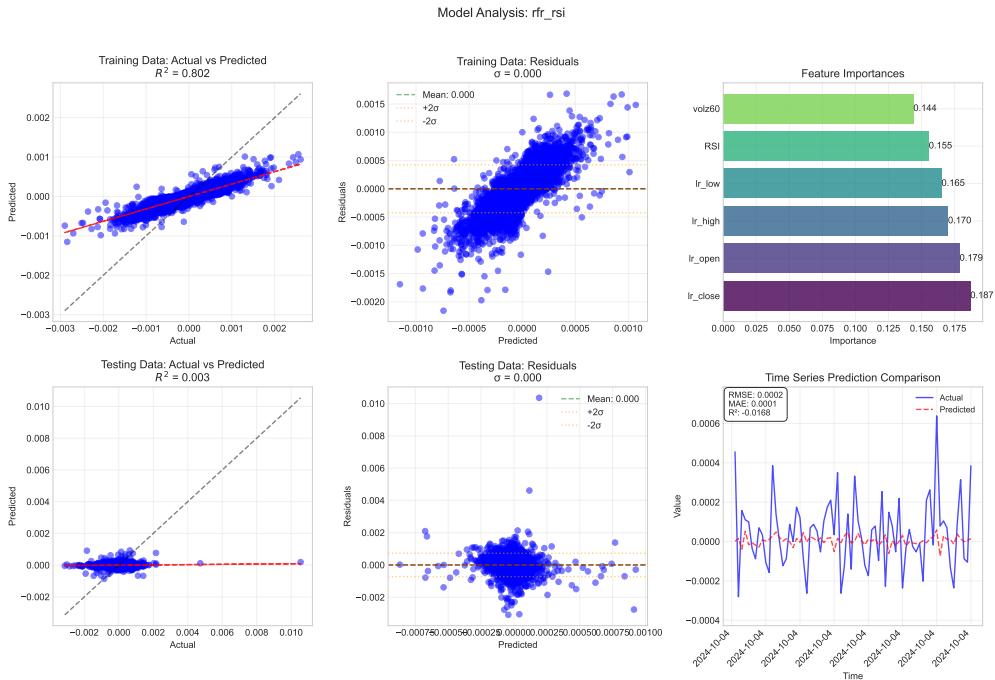


Figure 8: Feature importance distribution for the RFR model including RSI.

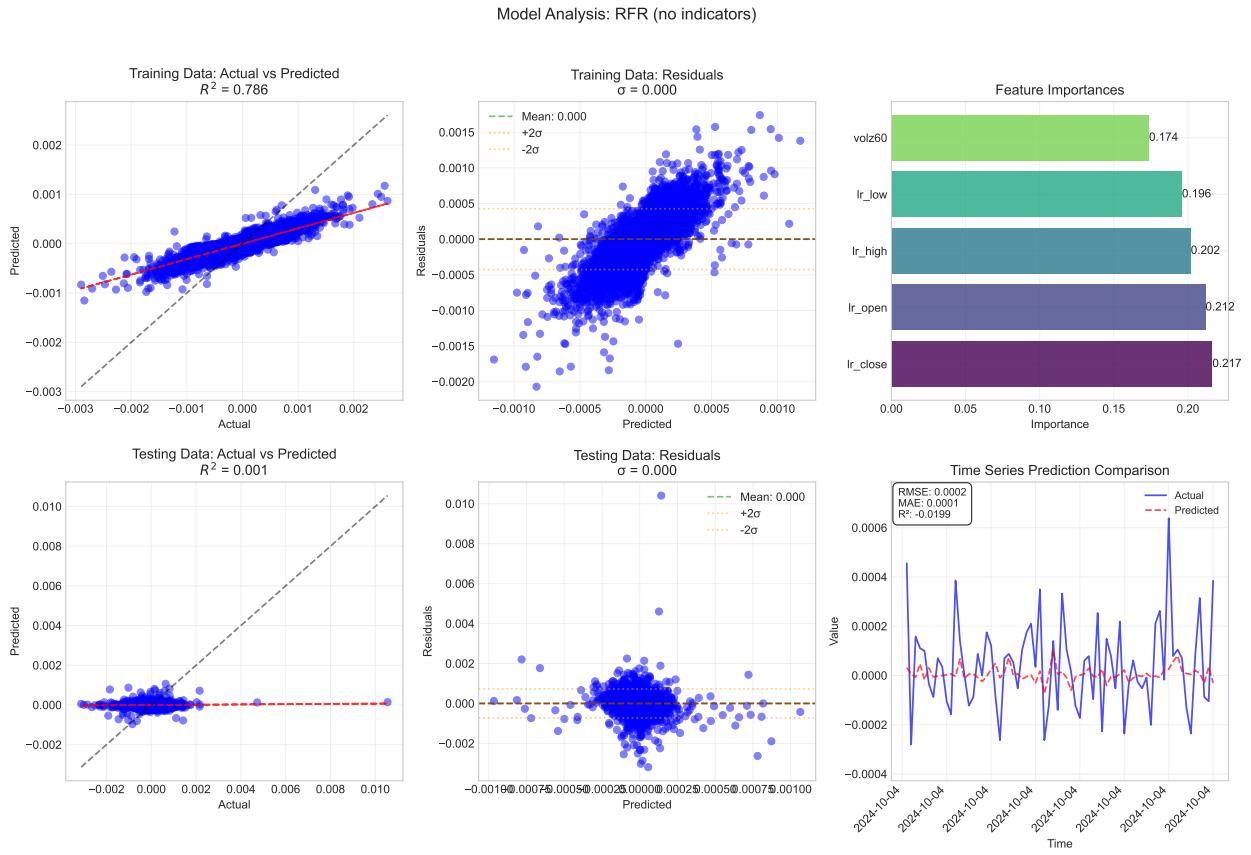


Figure 9: Residual analysis for the base RFR model: actual vs. predicted returns and residual distributions during training and testing.

alize, indicating that high-frequency signals may be overshadowed by market noise and short-term volatility.

5 Conclusion

This study investigated the integration of technical indicators into Random Forest Regression (RFR) models for high-frequency stock price prediction, emphasizing both predictive accuracy and risk-adjusted performance. Using minute-level SPY data, we systematically evaluated a range of technical indicators, including Bollinger Bands, Exponential Moving Averages (EMA), and Fibonacci retracements, to assess their contributions to the performance of the model under volatile market conditions. The choice of SPY, a highly liquid and representative market proxy, ensures that our findings retain their significance for broader high-frequency trading applications.

Our results indicate that while technical indicators enhance certain risk-adjusted metrics, such as the Rachev and gains–loss ratios, their contribution to out-of-sample predictive accuracy remains limited. A feature importance analysis consistently highlighted the dominance of primary price-based features (e.g., opening, closing, and high prices) over derived technical indicators. Hybrid strategies incorporating multiple indicators demonstrated slight improvements in managing tail risks but failed to outperform the buy-and-hold benchmark in terms of returns. These findings suggest that traditional technical indicators may have diminishing predictive value in modern high-frequency markets, where price discovery is driven primarily by raw price movements rather than widely recognized indicators.

Beyond predictive accuracy, this study advances the field by integrating advanced risk–reward measures to evaluate the practical viability of trading strategies based on machine learning (ML). While past research has focused predominantly on return maximization, our results emphasize the trade-offs between risk management and profitability. The observed difficulties in generalization, where models exhibit strong in-sample performance but deteriorate significantly in out-of-sample testing, highlight the need for parsimonious modeling approaches that prioritize robustness over complexity. This aligns with the existing literature on ML in financial markets, which finds overfitting to be a fundamental limitation in high-frequency trading applications.

From a theoretical standpoint, our findings provide insights into market efficiency and the feasibility of exploiting short-term price inefficiencies. While the inability of our models to consistently generate excess returns aligns with the weak form of the Efficient Market Hypothesis (EMH), the ability of certain indicator-augmented strategies to maintain stable risk–reward profiles suggests that transient inefficiencies may persist under specific market conditions. These results contribute to ongoing discussions on the microstructures of the market and the role of ML in financial decision-making.

Several challenges remain, including overfitting, the need for adaptive modeling techniques, and the computational costs associated with complex hybrid strategies. Future research should explore dynamic, regime-aware models capable of adjusting to evolving market conditions while maintaining their computational efficiency. Incorporating sources of alternative data, such as sentiment analysis and order book dynamics, could further enhance the predictive performance and provide deeper insights into price formation mechanisms.

From a practitioner’s perspective, this study highlights the importance of balancing interpretability, computational feasibility, and predictive power in the deployment of ML models for high-frequency trading. While RFR-based strategies may not be optimal for maximizing absolute returns, their ability to manage tail risks and provide interpretable outputs means they can be valuable tools for risk-aware trading strategies. Furthermore, technical indicators, such as Fibonacci

retracement and the Ichimoku Cloud, despite their limited predictive power, may still have some practical utility due to their alignment with intuitive trading heuristics.

In conclusion, this study contributes to the growing body of literature on ML in financial markets by providing a nuanced assessment of the role of technical indicators in high-frequency trading. While traditional indicators may have limited standalone predictive value, their integration within a structured risk-aware framework offers insights into market behavior and portfolio risk management. Future research should focus on adaptive hybrid approaches that address the challenges to generalization, leverage sources of alternative data, and optimize computational efficiency, to enhance the practical applicability of ML in modern financial markets.

6 Code Availability

The implementation code for this study is available at https://github.com/akashdeepo/ML_TI_RFR. The repository includes the core implementation files, `stockdata.py` for data processing and technical indicators, `pred_rfr.py` for the Random Forest model, `simulate_trading.py` for trading simulation, and `metrics.py` for performance evaluation.

The implementation uses the following Python libraries:

- `scikit-learn`: Random Forest implementation with `RandomForestRegressor`
- `pandas` and `numpy`: Data manipulation and numerical computations
- `matplotlib` and `seaborn`: Visualization and plotting
- `logging`: Comprehensive logging for debugging and tracking
- Custom modules:
 - Technical indicator computation
 - Trading simulation with position sizing and turnover constraints
 - Risk-reward ratio calculations including Rachev and Modified Rachev ratios

The implementation emphasizes computational efficiency and real-time processing capabilities, with particular attention to high-frequency trading considerations. The complete implementation requires access to minute-level SPY data through a Bloomberg Terminal subscription. Users wishing to replicate this study should have appropriate Bloomberg Terminal access and the necessary subscriptions. The code is provided as is under the MIT license, with the understanding that data acquisition and licensing compliance are the user's responsibility.

7 Future Work

While this study provides valuable insights into the role of technical indicators in high-frequency stock price prediction, several avenues remain open for further research. A key limitation of this study is its focus on a single asset, the SPY. Although the SPY was chosen for its high liquidity and broad market representation, future research should extend this analysis to multiple assets or multi-asset portfolios to evaluate the generalizability of the findings. Expanding the study to diverse asset classes, such as commodities, fixed-income securities, and cryptocurrencies, would provide a deeper understanding of how technical indicators interact with varying market structures, liquidity conditions, and volatility regimes.

Another important direction is the integration of additional data sources to enhance the predictive performance and risk assessment. Order book dynamics, sentiment analysis from financial news and social media, and alternative data sources such as macroeconomic indicators, could improve the feature selection and provide more context for trading decisions. Investigating how these factors influence the performance of a model in high-frequency environments may yield more robust trading strategies.

Further, advances in deep learning architectures present an opportunity to capture complex sequential dependencies in high-frequency financial data. Future studies should explore models such as Long Short-Term Memory (LSTM) networks and Transformer-based architectures, which have demonstrated strong performance in time-series forecasting tasks. Additionally, comparisons with alternative ML techniques, such as gradient boosting methods or hybrid ensemble models, could provide insights into the optimal modeling approaches for different market conditions.

Finally, the challenges to practical implementation must be addressed to ensure the viability of ML-driven trading strategies in real-world applications. Future research should explore the development of adaptive frameworks that dynamically adjust to evolving market regimes while incorporating real-world constraints, such as transaction costs, latency, and execution risks. The integration of reinforcement learning techniques so as to optimize the execution of trades and risk management strategies could further enhance the applicability of ML models in high-frequency trading.

By pursuing these research directions, future studies can contribute to the development of more resilient, interpretable, and efficient ML models for financial markets, ultimately bridging the gap between theoretical advances and practical deployment in trading environments.

References

- Abrol, S., Chesir, B., Mehta, N., & Ziegler, R. (2016). High frequency trading and US stock market microstructure: A study of interactions between complexities, risks and strategies residing in US equity market microstructure. *Financial Markets, Institutions & Instruments*, 25(2):107–165.
- Agrawal, M., Khan, A.U., & Shukla, P.K. (2019). Stock price prediction using technical indicators: A predictive model using optimal deep learning. *Learning*, 6(2):7.
- Akyildirim, E., Bariviera, A.F., Nguyen, D.K., & Sensoy, A. (2022). Forecasting high-frequency stock returns: a comparison of alternative methods. *Annals of Operations Research*, 313(2):639–690.
- Akyildirim, E., Cepni, O., Corbet, S., & Uddin, G.S. (2023). Forecasting mid-price movement of Bitcoin futures using machine learning. *Annals of Operations Research*, 330(1):553–584.
- Aldridge, I. (2013). *High-frequency trading: A practical guide to algorithmic strategies and trading systems*. John Wiley & Sons.
- Alexander, G.J., & Baptista, A.M. (2003). Portfolio performance evaluation using value at risk. *The Journal of Portfolio Management*, 29(4):93–102.
- Arifovic, J., He, X., & Wei, L. (2022). Machine learning and speed in high-frequency trading. *Journal of Economic Dynamics and Control*, 139:104438.
- Artzner, P., Delbaen, F., Eber, J.M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3):203–228.

- Aumann, R.J., & Serrano, R. (2008). An economic index of riskiness. *Journal of Political Economy*, 116(5):810–836.
- Ayala, J., García-Torres, M., Noguera, J.L.V., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. *Knowledge-Based Systems*, 225:107119.
- Ballings, M., Van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056.
- Barberis, N. (2003). A Survey of Behavioral Finance. *Handbook of the Economics of Finance*, 1.
- Bloomberg L.P. (2024). *Bloomberg Terminal*. Accessed: 09-2024.
- Bollinger, J. (2002). *Bollinger on Bollinger Bands*. McGraw-Hill, New York.
- Box, G.E.P., Jenkins, G.M., & Reinsel, G.C. (1994). Time series analysis: forecasting and control, Prentice Hall. *Englewood Cliffs*.
- Box, G.E.P., Jenkins, G.M., Reinsel, G.C., & Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., et al. (unpublished). API design for machine learning software: Experiences from the scikit-learn project. *arXiv:1309.0238*.
- Chandar, S.K. (2022). Convolutional neural network for stock trading using technical indicators. *Automated Software Engineering*, 29(1):16.
- Cheridito, P., & Kromer, E. (Forthcoming). Reward-risk ratios. *Journal of Investment Strategies*.
- Dash, R., & Dash, P.K. (2016). A hybrid stock trading framework integrating technical analysis with machine learning techniques. *The Journal of Finance and Data Science*, 2(1):42–57.
- Deep, A. (2023). A Multifactor Analysis Model for Stock Market Prediction. *International Journal of Computer Science and Telecommunications*, 14(1).
- Deep, A. (2024). Advanced Financial Market Forecasting: Integrating Monte Carlo Simulations with Ensemble Machine Learning Models. *Quantitative Finance and Economics*, 8(2).
- Derbentsev, V., Matviychuk, A., Datsenko, N., Bezkorovainyi, V., & Azaryan, A.A. (2020). Machine learning approaches for financial time series forecasting. In *Proceedings of the Selected Papers of the Special Edition of International Conference on Monitoring, Modeling & Management of Emergent Economy, Odessa, Ukraine, July 13–18, 2020*, pages 434–450. CEUR Workshop Proceedings.
- Easley, D., López de Prado, M., O’Hara, M., & Zhang, Z. (2021). Microstructure in the machine age. *The Review of Financial Studies*, 34(7):3316–3363.
- El Hajj, M., & Hammoud, J. (2023). Unveiling the influence of artificial intelligence and machine learning on financial markets: A comprehensive analysis of AI applications in trading, risk management, and financial operations. *Journal of Risk and Financial Management*, 16(10):434.

- Engle, R.F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007.
- Engle, R.F., & Russell, J.R. (2004). Analysis of high frequency financial data. *Handbook of Financial Econometrics*.
- Falck, A., Rej, A., & Thesmar, D. (2022). When do systematic strategies decay? *Quantitative Finance*, 22(11):1955–1969.
- Fama, E.F. (1970). Efficient capital markets. *Journal of Finance*, 25(2):383–417.
- Ferrouhi, E.M., & Bouabdallaoui, I. (2024). A comparative study of ensemble learning algorithms for high-frequency trading. *Scientific African*, 24:e02161.
- Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.
- Frattini, A., Bianchini, I., Garzonio, A., & Mercuri, L. (2022). Financial technical indicator and algorithmic trading strategy based on machine learning and alternative data. *Risks*, 10(12):225.
- Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). Random forests. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, pages 587–604. Springer, Berlin.
- Ho, T.K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE.
- Hossain, M.A., Karim, R., Thulasiram, R., Bruce, N.D.B., & Wang, Y. (2018). Hybrid deep learning model for stock price prediction. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1837–1844. IEEE.
- Hsu, M.W., Lessmann, S., Sung, M.C., Ma, T., & Johnson, J.E.V. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications*, 61:215–234.
- Kearns, M., & Nevyvaka, Y. (2013). *Machine learning for market microstructure and high frequency trading*, volume 72. Risk Books: London, UK.
- Lambert, D.R. (1983). Commodity Channel Index: Tool for trading cyclic trends. *Technical Analysis of Stocks & Commodities*, 1:47.
- Letteri, I. (2023). VolTS: A Volatility-based Trading System to forecast Stock Markets Trend using Statistics and Machine Learning. *arXiv preprint arXiv:2307.13422*.
- Li, A.W., & Bastos, G.S. (2020). Stock market forecasting using deep learning and technical analysis: a systematic review. *IEEE Access*, 8:185232–185242.
- Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.
- Lo, A.W., Mamaysky, H., & Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *The Journal of Finance*, 55(4):1705–1765.

- McGroarty, F., Booth, A., Gerding, E., & Chinthalapati, V.L.R. (2019). High frequency trading strategies, market fragility and price spikes: An agent based model perspective. *Annals of Operations Research*, 282(1):217–244.
- McNeil, A.J., Frey, R., & Embrechts, P. (2015). *Quantitative risk management: concepts, techniques and tools-revised edition*. Princeton University Press.
- Mirete-Ferrer, P.M., Garcia-Garcia, A., Baixauli-Soler, J.S., & Prats, M.A. (2022). A review on machine learning for asset management. *Risks*, 10(4):84.
- Moskowitz, T.J., Ooi, Y.H., & Pedersen, L.H. (2012). Time series momentum. *Journal of Financial Economics*, 104(2):228–250.
- Murphy, J.J. (1999). *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Penguin.
- O'Hara, M. (2015). High frequency market microstructure. *Journal of Financial Economics*, 116(2):257–270.
- Pardeshi, Y.K., & Kale, P. (2021). Technical analysis indicators in stock market using machine learning: A comparative analysis. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.
- Pástor, L., & Stambaugh, R.F. (2003). Liquidity risk and expected stock returns. *Journal of Political Economy*, 111(3):642–685.
- Patel, M. (2010). *Trading with Ichimoku Clouds: The Essential Guide to Ichimoku Kinko Hyo Technical Analysis*. John Wiley & Sons.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268.
- Peng, Y., Albuquerque, P.H.M., Kimura, H., & Saavedra, C.A.P.B. (2021). Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. *Machine Learning with Applications*, 5:100060.
- Rundo, F., Trenta, F., Di Stallo, A.L., & Battiatto, S. (2019). Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574.
- Salkar, T., Shinde, A., Tamhankar, N., & Bhagat, N. (2021). Algorithmic trading using technical indicators. In *2021 International Conference on Communication information and Computing Technology (ICCICT)*, pages 1–6. IEEE.
- Shleifer, A., & Vishny, R.W. (1997). The limits of arbitrage. *The Journal of Finance*, 52(1):35–55.
- Shreve, S.E., et al. (2004). *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer.
- Stoyanov, S.V., Rachev, S.T., & Fabozzi, F.J. (2007). Optimal Financial Portfolios. *Applied Mathematical Finance*, 14(5):401–436.

- Wang, J., Sun, T., Liu, B., Cao, Y., & Wang, D. (2018). Financial markets prediction with deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 97–104. IEEE.
- Wang, X., Wang, Y., Weng, B., & Vinel, A. (2020). Stock2Vec: a hybrid deep learning framework for stock market prediction with representation learning and temporal convolutional network. *arXiv preprint arXiv:2010.01197*.
- Wilder, J.W. (1978). *New Concepts in Technical Trading Systems*. Greensboro, NC.
- Zanc, R., Cioara, T., & Anghel, I. (2019). Forecasting financial markets using deep learning. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 459–466. IEEE.
- Zhang, G., Patuwo, B.E., & Hu, M.Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1):35–62.
- Zhang, F. (2010). High-frequency trading, stock volatility, and price discovery. *Available at SSRN 1691679*.