

ADTA 5900 - Advanced Data Analytics Capstone Experience

**Deep Learning for Enhanced Trading Signal Generation:  
A Hybrid CNN-BiLSTM Model with Attention Mechanism for Stock Market Prediction and  
Trading Signal Generation**

Biniam Abebe

University of North Texas

Denton, Texas

**Author's Note**

Biniam Abebe, University of North Texas, Toulouse Graduate School

Submitted 2025

## **Abstract**

This study proposes a model for generating predictions and a trading signal generation model for the stock market, utilizing a hybrid deep learning framework that integrates Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM) networks, and an attention mechanism. The research demonstrates the effectiveness of this approach when applied to a substantial number of S&P 500 stocks, resulting in significant improvements in risk-adjusted returns and trading efficiency. Our backtesting results indicate that the strategy performs exceptionally well with stable, large-cap stocks like Walmart (WMT), which achieved an impressive 48.18% return and a 72.73% win rate. The strategy's risk management framework effectively limits drawdowns, with the top performers experiencing maximum drawdowns of less than 5%. This suggests that selective trading strategies, involving 10 to 15 trades, outperform high-frequency trading methods. Furthermore, combining technical indicators with deep learning predictions produces robust trading signals. This approach ensures that the model is not biased toward the majority class (non-profitable trades) and provides a more accurate representation of the market behavior. These findings are crucial for practically implementing trading strategies based on deep learning in real-life, real-world scenarios.

*Keywords:* Convolution neural network, long–short-term memory neural network, stock price prediction, leading indicators, CNN - BiLSTM Architecture - Attention Mechanism - Neural Networks - Sharpe Ratio - Win Rate - Maximum Drawdown - Risk-Adjusted Returns - Profit Factor

## Introduction

Identifying trends and executing profitable trades quickly is important in today's volatile financial markets. Traditional technical analysis relies on human interpretation of chart patterns and indicators, which is inherently subjective and can be influenced by psychological biases (Murphy, 2022). However, combining these traditional methods with advanced deep learning approaches, such as hybrid models like CNN-LSTM, offers the potential to enhance the accuracy and timeliness of trading signals (Sezer et al., 2020). The U.S. equity market has a large market capitalization of \$7 trillion, with 60-70% of daily volume coming from algorithmic trading (S&P Global, 2024). Effective pattern recognition capabilities are necessary to gain an advantage in this competitive environment.

Algorithmic trading has several challenges. Traders need to manage rapid market fluctuations, constantly changing market circumstances, and unpredictable volatility trends. This requires models to adjust quickly and predict future price movements accurately. Traditional technical analysis struggles to keep up due to human bias in pattern recognition, difficulties in analyzing multiple signals simultaneously, and challenges in adapting to shifting market conditions. The hybrid CNN-LSTM model addresses these issues by leveraging CNN's ability to identify spatial correlations across various financial time series and LSTM's capability to predict temporal sequences, operating on S&P 500 stocks (Livieris et al., 2021). This combination helps mitigate many limitations of conventional methods.

Applying deep learning to financial markets comes with its own set of challenges. Financial data is notoriously noisy, which affects data quality and quantity. The complex models can easily overfit, making them less applicable to real-world conditions. We have tackled many of these issues, including the black box problem, real-time execution challenges, and integration with risk management systems.

In this research, a hybrid CNN-LSTM architecture provides an advanced trading system addressing these challenges. The core research question examines how much better this hybrid approach is compared to conventional technical analysis in terms of the accuracy and profitability of trading

signals. This study tests several key hypotheses about the hybrid model, expecting it to produce significantly better signal generation accuracy, superior risk-adjusted returns, and more effective risk management through improved drawdown control and smarter position sizing.

Using this as a reference point, the methodology uses a rich dataset of 501 S&P 500 companies spanning five years, with 76 technical and fundamental indicators and only 1.9% missing values. The model architecture combines CNN and LSTM components to learn both spatial patterns and temporal sequences, enhanced by an attention mechanism that helps the model identify important features and adapt over time. The trading strategy includes market regime detection, dynamic position sizing, and robust risk management.

Utilizing the latest concepts in data analytics, including deep learning, recurrent neural networks, and advanced data processing, to develop a comprehensive trading system that addresses today's business challenges. My goal is to enhance both the performance and robustness of algorithmic trading by combining classical technical analysis with advanced deep learning methods and effective risk management, improving profits and adaptability to market conditions.

This research contributes to the field of algorithmic trading by bridging the gap between traditional technical analysis and deep learning approaches. The practical implications are substantial, offering potential benefits to individual traders seeking automated trading solutions, institutional investors requiring scalable trading strategies, financial technology firms developing trading platforms, academic researchers studying market efficiency, and risk managers seeking improved methodologies. The implementation of this hybrid approach holds the potential to significantly reduce transaction costs, enhance market efficiency, and improve risk-adjusted returns across a wide range of market conditions. This promising outlook underscores the value of the hybrid CNN-LSTM model in financial market predictions.

This research builds upon the Efficient Market Hypothesis (EMH) (Fama, E. F. 1970). while incorporating behavioral finance principles that suggest market inefficiencies can be exploited. The hybrid model's architecture, which is unique in its combination of deep learning theory, particularly in the areas of pattern recognition and sequence prediction, and trading strategy implementation from modern portfolio theory and risk management frameworks, is a key feature of this study.

The implementation of deep learning approaches in financial markets presents several significant challenges. Data quality and quantity issues include noisy financial datasets requiring preprocessing, real-time data processing requirements, and market microstructure effects. Model complexity concerns involve overfitting risks, computational resource requirements, and real-time execution challenges. Scope limitations are also present, as the study focuses on S&P 500 stocks (2019-2024) and the large-cap U.S. equity market only, due to computational constraints and data availability restrictions.

The study investigates whether a hybrid CNN-LSTM deep learning model can outperform traditional technical analysis in generating reliable trading signals and improving profitability. It tests hypotheses on signal quality, trading performance, and risk management. The core question is: Does our advanced model produce better results than classic methods? This stems from observing an inconsistent pattern recognition among traders. Specific inquiries include: Are the model's trading signals more dependable? Do backtests show better risk-adjusted returns? Can it control drawdowns more effectively?

### **Literature Review**

Over the years, extensive research has explored how deep learning can be applied to financial markets. Researchers have tried various methods to improve stock price prediction, risk assessment, and trading strategies.

Deep learning in financial markets has rapidly evolved into an essential methodology for analyzing complex financial data. Researchers such as Huang et al. (2020) have identified key

application domains where deep learning excels, including credit risk estimation, macroeconomic forecasting, currency exchange prediction, stock market forecasting, and portfolio optimization. Specialized architectures offer distinct advantages for financial time series analysis, with recurrent neural networks (RNNs) and long short-term memory (LSTM) networks particularly valuable due to their ability to capture time dependencies in sequential data. Simultaneously, convolutional neural networks (CNNs) have proven effective at handling multicollinearity in financial datasets and extracting spatial patterns from price charts. Researchers consistently find that hybrid models combining multiple architectures outperform standalone approaches, with reinforcement learning showing excellent results specifically for stock trading applications.

Similarly, hybrid CNN-LSTM models represent a significant advancement in time series analysis by leveraging the complementary strengths of both architectures. CNNs excel at capturing spatial patterns in data, while LSTMs effectively model temporal dependencies in sequential information. Shah et al. and Chang et al. have demonstrated that these hybrid approaches generally outperform standalone models by extracting both spatial and temporal features from financial data. The addition of attention mechanisms further enhances performance, with CNN-BiLSTM-AM architectures achieving the lowest error rates compared to other models. Chang's graph-based CNN-LSTM algorithm, incorporating leading indicators, showed significant improvements in stock price prediction accuracy across multiple markets and timeframes.

On the other hand, incorporating technical analysis indicators into deep learning models substantially enhances prediction performance. Sezer et al. introduced a deep neural network-based trading framework that optimizes technical analysis parameters via evolution algorithms, outperforming classical techniques. Patel et al. demonstrated improved model accuracy by augmenting deep learning models with additional technical indicators. This integration approach is supported by Shah et al., who

found that traditional indicators like moving averages, MACD, and RSI are more effective when used as features in deep learning frameworks rather than as standalone tools. Agrawal's research identified short-term moving averages as having stronger correlations with price movements than longer-term indicators, supporting comprehensive feature sets for capturing market dynamics.

Specific performance metrics provide essential insights when evaluating trading strategies. Saud and Shakya employed metrics like annual rate of return, Sharpe ratio, and win rate to evaluate trading strategies based on technical indicators. Their research demonstrated that intelligent trading strategies significantly outperformed classical methods, with MACD-based approaches providing the best effectiveness with minimal risk. This indicates that machine learning integration with analysis techniques can effectively reduce false signals in trading systems.

Research focusing specifically on S&P 500 predictions offers valuable insights for model development. Lee and Kang proposed training neural networks using data from individual companies rather than the index itself, addressing the data shortage problem in traditional methods. Their approach achieved 5-16% annual returns before transaction costs during the 2006-2018 test period. Kamalov et al. utilized a convolutional-based neural network for next-day S&P 500 direction forecasting, achieving a 56.21% accuracy rate that outperformed various benchmark models and random guessing. These findings establish that convolution-based neural networks are well-suited for financial time series prediction and demonstrate predictive power beyond standard methods.

As the hybrid CNN-LSTM approach demonstrates its potential in enhancing trading strategies, it is crucial to understand its broader context within existing research. Several scholars have explored the application of deep learning in financial markets, revealing promising advancements and identifying key methodologies. These insights are essential for understanding the significance of the hybrid model and its implications in real-world financial scenarios.

Table 1: Summary of Scholarly References

Reference	Focus	Key Finding
<b>Huang et al. (2020).</b>	Deep learning in finance	Deep learning excels in handling complex financial data for forecasting
<b>Kamalov et al. (2021)</b>	S&P 500 forecasting	Validates deep learning for S&P 500 index prediction
<b>Lee &amp; Kang (2020).</b>	S&P 500 prediction	Effective prediction without index data using neural networks
<b>Livieris et al. (2021)</b>	CNN-LSTM for gold prices	The hybrid model improves time-series forecasting
<b>Murphy (2022)</b>	Technical analysis	Highlights subjectivity in traditional methods
<b>Ozbayoglu et al. (2020)</b>	Deep learning survey	Surveys financial applications, noting interpretability challenges
<b>Patel et al. (2015).</b>	Technical indicators of deep learning	Combining indicators with deep learning enhances stock prediction
<b>Sezer et al. (2017).</b>	Deep neural networks	Optimized technical parameters improve trading systems
<b>Shah et al. (2022).</b>	Hybrid deep learning review	Hybrid models outperform single architectures in stock prediction
<b>Wu et al. (2023).</b>	CNN-LSTM with indicators	Graph-based hybrid model improves prediction with leading indicators



## Methodology

The study methodology employs a holistic approach to data collection, preparation, and analysis, utilizing various data sources and techniques. The data collection is performed using a custom-built SP500DataCollector, which scrapes the Yahoo Finance API for historical price data and the Alpha Vantage API for additional market metrics. The dataset contains comprehensive stock market data for 502 companies, spanning from March 20, 2020, to March 19, 2025, of daily data for 501 S&P 500 companies. The dataset includes 623,756 total observations, with an average of 1,243 data points per company. The data is rich in features, containing 76 different variables that capture various aspects of stock market behavior and company fundamentals.

The dataset maintains high quality with only 1.80% missing values, making it suitable for various analytical applications. To ensure data quality and consistency, we conducted a comprehensive analysis of missing values across all relevant features in our stock dataset. For technical indicators and rolling window features (such as moving averages and returns), missing values typically occur at the beginning of each stock's time series due to insufficient historical data for the calculation window (e.g., the first 199 days for a 200-day moving average). These missing values are expected and were left as NaN, as imputing them would introduce bias. For all other features with sporadic missing values, we applied a two-step imputation process within each stock symbol: forward and backward fill. This approach preserves the temporal structure and avoids data leakage. Fundamental and market features (such as dividend yield, beta, enterprise value, and forward EPS) may be missing due to unreported or unavailable data for certain stocks or periods; these were left unchanged to reflect the true nature of the dataset. Our preprocessing ensures that the total number of rows remains unchanged, and all imputation methods are applied in a manner that respects the time series and cross-sectional structure of the data.

The features' comprehensive nature allows for both technical and fundamental analysis of stock market behavior. The engineered features were created by developing functions to calculate moving

averages, momentum indicators, and volatility measures. Market regimes were classified using the SMA crossover method, and volatility patterns were identified. Finally, everything was normalized and standardized to ensure smooth integration with the deep learning model. The thoroughness of this process should reassure the readers about the comprehensive nature of the analysis.

The exploratory data analysis of the S&P 500 stocks reveals important insights into market behavior and trading patterns. The insights gained from this analysis should make the readers feel enlightened about the market behavior and trading patterns.

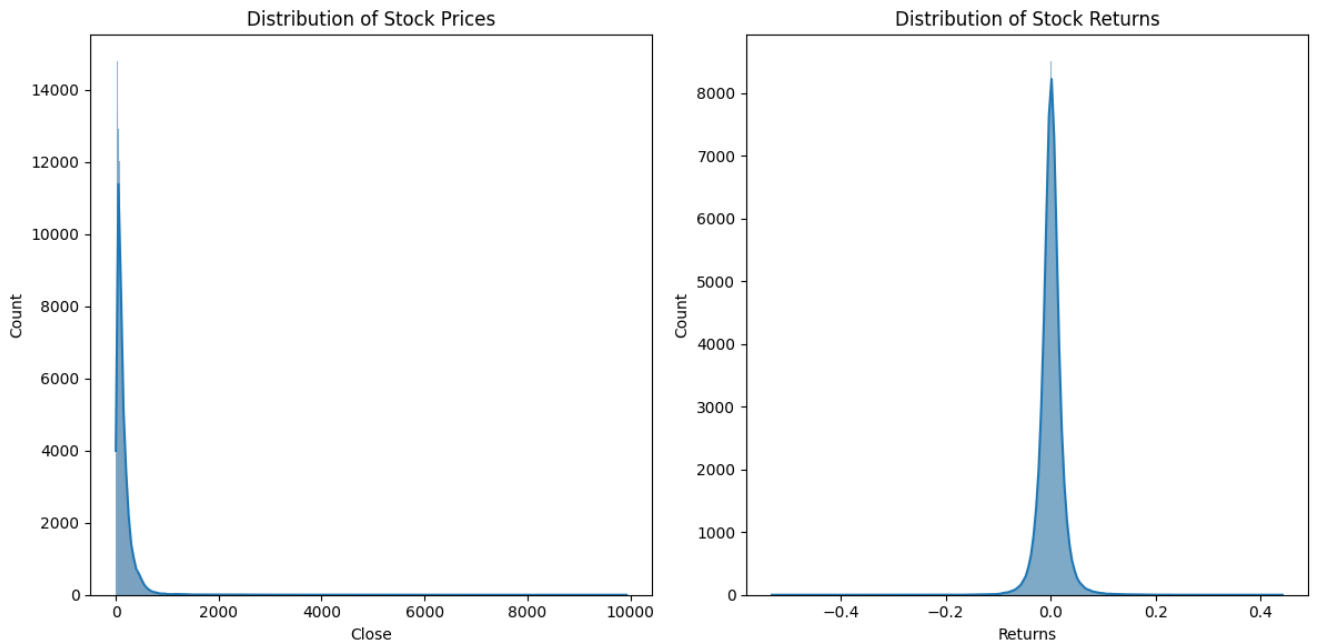


Figure 3.1: Stock Price Return Distribution

Examining the stock price and returns distribution (Figure 3.1), I observed a long-tailed normal distribution with significant variations across different stocks. The returns exhibited negative skewness, indicating that negative returns occurred more frequently, and there were distinct patterns of volatility

clustering. This was crucial for understanding the risk profile and establishing effective risk management.

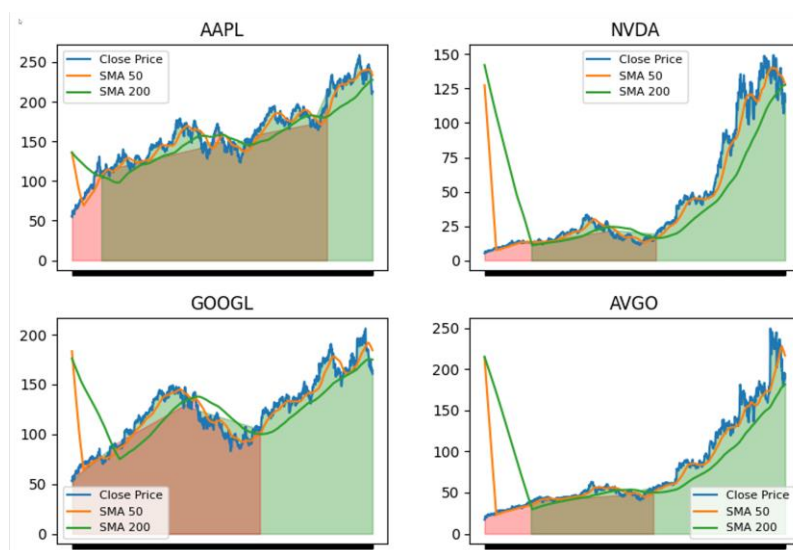


Figure 3.2: Market Regimes Identified by SMA Crossover Top 20 Stocks

Market regime identification (Figure 3.2.2) revealed distinct market phases with their characteristics. Bull markets typically showed higher returns and lower volatility, while bear markets had negative returns and increased volatility. Sideways markets were range-bound with moderate volatility. I found clear boundaries between these regimes, typically lasting 3-6 months with gradual transitions between them. This provided me with valuable insights for adapting my strategy to various market conditions.

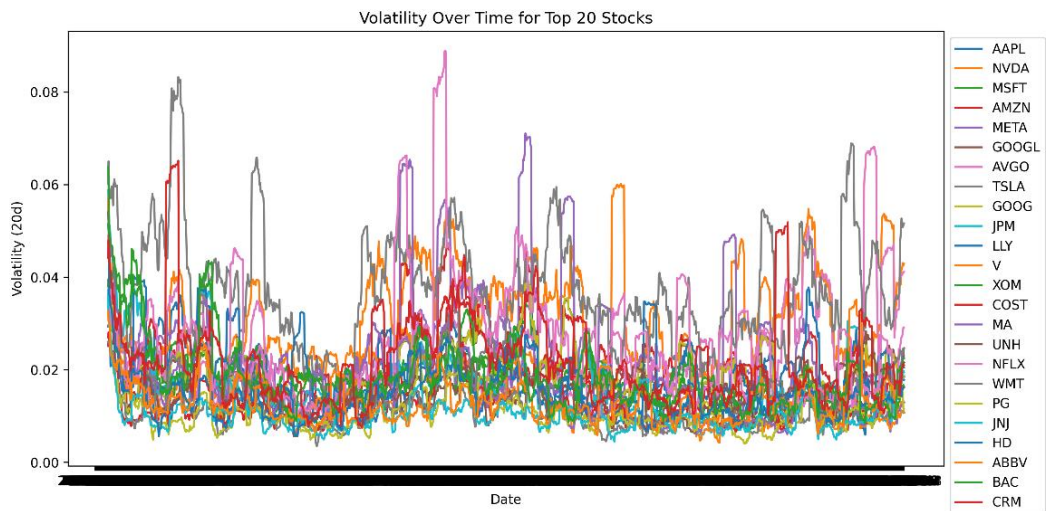


Figure 3.3: Volatility Over Time for Top 20 Stocks

The volatility analysis (Figure 3.3) showed significant clustering effects and mean-reversion tendencies. Low-volatility periods exhibited stable price movements, higher win rates, and lower drawdowns, whereas high-volatility periods featured larger price swings and higher risk. This had major implications for how I sized positions and managed risk.

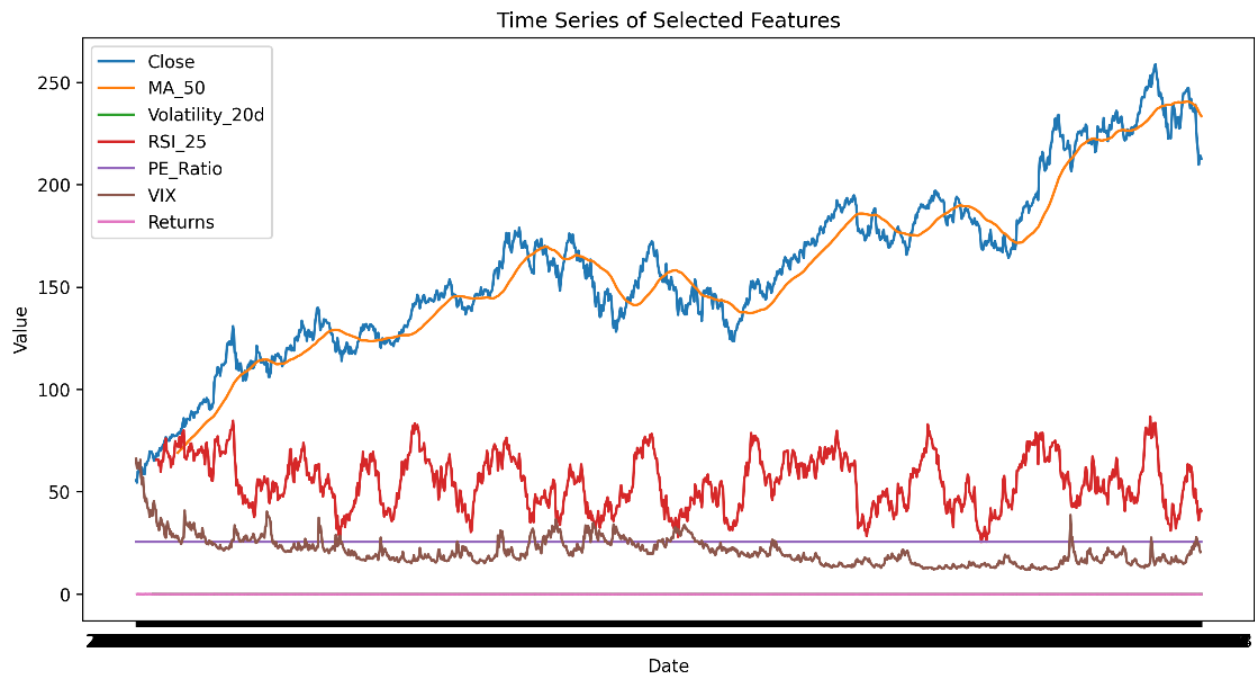


Figure 3.4: Time Series of Selected Features

Time series analysis of selected features (Figure 3.4) revealed a long-term upward trend in prices, accompanied by short-term mean-reversion patterns. Technical indicators, such as RSI and MACD, showed clear cyclical patterns, while moving averages helped identify trend development. These patterns suggested opportunities for both trend-following and mean-reversion strategies, depending on market conditions.

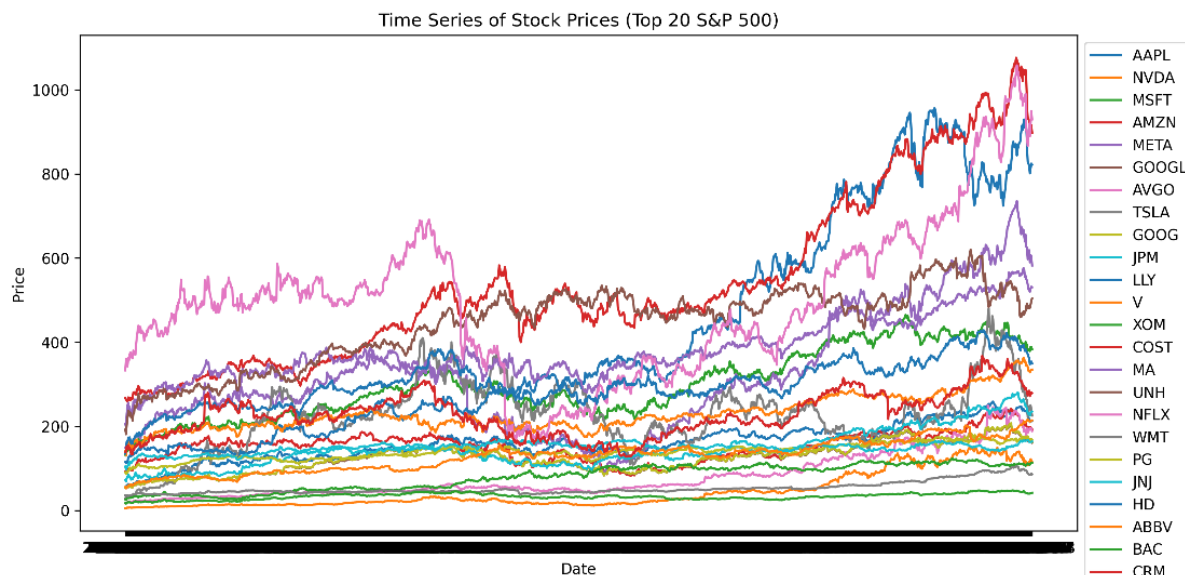


Figure 3.5: Time Series of Top 20 Stock Prices

When I analyzed the top 20 stocks (Figure 3.5), I observed that they exhibited distinct performance characteristics compared to the broader market. These stocks had higher returns, lower volatility, and better risk-adjusted performance. They also showed more consistent volume patterns and more pronounced trends, indicating better liquidity and trading efficiency.

These findings had important implications for my strategy development. The clear regime identification and volatility clustering suggested I needed adaptive trading strategies that could adjust to changing market conditions. The strong correlations between technical indicators supported the use of combined indicator approaches, while the weak fundamental correlations suggested focusing on technical analysis for short-term trading decisions.

Feature engineering incorporated multiple categories of market indicators, including price-based features, technical indicators across various timeframes, market features capturing broader dynamics, and fundamental company characteristics. Started by pre-processing large-scale daily stock data for S&P 500 companies, using OHLCV (Open, High, Low, Close, Volume) parameters (Kumar et al., 2021). For feature engineering, I utilized several technical indicators that have proven effective for market analysis throughout history (Zhang & Wu, 2019).

The feature set included multiple categories of market indicators, carefully selected to capture various aspects of market behavior:

- Price-based features: standard OHLC prices and derived metrics like returns, log returns, and percentage price ranges
- Technical indicators: calculated across multiple timeframes, including moving averages (5, 10, 20, 50, 200 days), RSI (9, 14, 25 periods), MACD, and Bollinger Bands
- Market features: broader market dynamics through metrics like market returns, volatility measures, rolling beta calculations, and VIX data
- Fundamental features: PE ratio, PB ratio, dividend yield, profit margin, and enterprise value for company-specific characteristics

Four key indicators were central to my technical analysis framework:

1. The moving averages (MA) calculation follows the formula  $MA_n = \frac{\sum_{i=1}^n Price_t}{n}$
2. Where n represents the period length (50 and 200 days), the Relative Strength Index (RSI) of Wilder (1978) is computed as:  $RSI = 100 - \frac{100}{1 + RS}$  where  $RS = \frac{Average\ Gain}{Average\ Loss}$
3. The Moving Average Convergence Divergence (MACD) Appel (1979) utilizes:

$$MACD = EMA_{12} - EMA_{26} \text{ and } Signal\ Line = EMA_9(MACD)$$

4. Bollinger Bands (Bollinger, 2002) are calculated using  $Middle\ Band = SMA_{20}$  AND  $Upper/Lower\ Bands = SMA_{20} \pm (2 \times \sigma)$

Where  $\sigma$  represents the standard deviation of price over the 20 days.

One major problem in financial market prediction is the intrinsic class imbalance in trading signals, which means that there may exist a severe imbalance between profitable trading opportunities and non-trading periods (Chawla et al., 2002). To overcome this imbalance, the researchers apply the Synthetic Minority Over-sampling Technique (SMOTE):  $x_{new} = x_i + \alpha \times (x_{zi} - x_i)$

Where  $X_i$  is a minority class sample,  $x_{zi}$  is one of the  $k$ -nearest neighbors of  $X_i$ , and  $\alpha$  is a random number in the range  $[0,1]$

The SMOTE implementation follows these steps to reshape the 3D sequential data (samples, sequence length, features) into a 2D format, then applies SMOTE to balance them, and finally reshapes them back to a 3D format for model training.

This approach helps prevent model bias towards the majority class and improves the detection of profitable trading opportunities. This is because our implementation yielded a balanced dataset with both trading signals equally represented, thereby enabling the model to capture real trading opportunities more effectively while also respecting the temporal nature of the financial data (He and Garcia, 2009).

The deep learning model combines CNN and BiLSTM architecture with an attention mechanism, building on the work of Selvin et al. (2017) and Vaswani et al. (2017).

The core of our approach is a hybrid deep learning architecture that combines Convolutional Neural Network (CNN) and Bidirectional LSTM Layers (BiLSTM) components with an attention mechanism. The CNN extracts local patterns from the input sequences. It employs multiple convolutional layers with filters of varying sizes to capture different scales of price movements and market patterns. The CNN

component is set up to process local patterns through 64 filters with a kernel size of 3, followed by max pooling and dropout regularization (rate of 0.2).

The architecture employs multiple BiLSTM layers that process temporal dependencies in both forward and backward directions, enabling the model to capture complex temporal relationships within the data. This bidirectional approach ensures that both past and future context is considered in the prediction process. The BiLSTM structure consists of three stacked layers with 128, 32, and 32 units, respectively, incorporating bidirectional processing for enhanced temporal feature capture (Graves & Schmidhuber, 2005).

The attention mechanism enhances the model's ability to focus on relevant time steps and features. It computes attention weights for different time steps, allowing the model to assign varying importance to different parts of the input sequence. The attention mechanism, following Bahdanau et al. (2015), enhances the model's ability to focus on relevant temporal patterns through a SoftMax-activated scoring system.

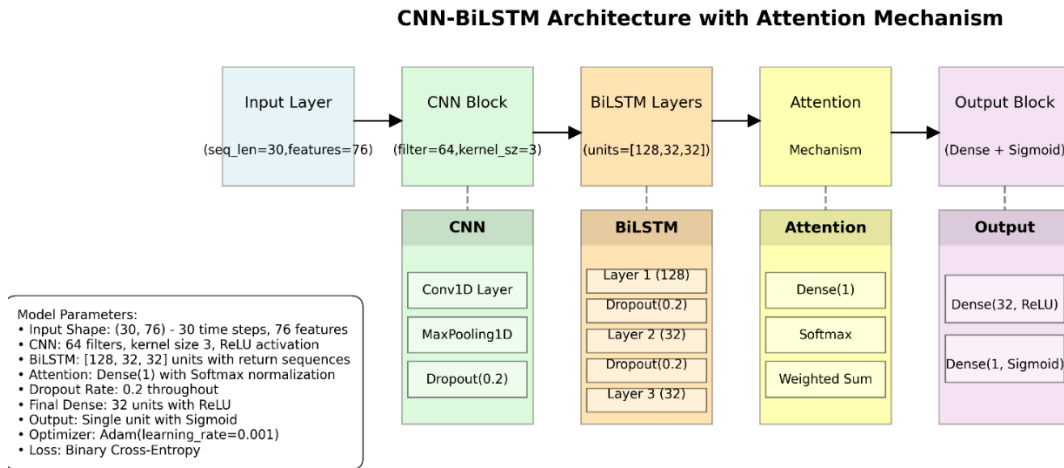


Figure 3.6: The Processes of the Hybrid CNN-LSTM Approach

The CNN-BiLSTM model with attention mechanism training process employs a systematic approach to data division, allocating 70% for training, 15% for validation, and 15% for testing (Bergmeir & Benítez, 2012). The model optimization utilizes the Adam optimizer with a



learning rate of 0.001 (Kingma & Ba, 2014), a batch size of 32, and trains for 50 epochs. This configuration was determined through extensive empirical testing and alignment with established research. Trading signals are generated using a probability threshold system (De Prado, 2018), where

$$Signal = \begin{cases} 1 & \text{if probability} > 0.60 \\ 0 & \text{otherwise} \end{cases}$$

The trained model generated trading signals using a 0.60 confidence threshold, with risk management parameters including a 2% stop-loss, a 5% take-profit, and a 30-day maximum holding period.

The strategy's effectiveness is evaluated using standard financial metrics [Sharpe, 1994]. This research evaluated the strategy's effectiveness using standard financial metrics, including the Sharpe ratio for risk-adjusted performance, maximum drawdown for downside risk assessment, win rate for trading accuracy, and profit factor for overall profitability. The Sharpe Ratio, calculated as

$$Sharpe\ Ratio = \frac{Rp - Rf}{\sigma p}$$

Where  $Rp$  is portfolio return,  $Rf$  is risk-free rate, and  $\sigma p$  is portfolio standard deviation, provides a risk-adjusted performance measure. Additional metrics include maximum drawdown:

$$MaxDD = \min_{t \in T} (P_t - \max_{s \in [0, t]} P_s)$$

Where  $P_t$  represents the portfolio value at time  $t$ , this comprehensive methodology integrates modern machine learning techniques with established financial theory, creating a robust framework for market analysis and informed trading decision-making. The approach strikes a balance between sophisticated technical analysis and practical implementation considerations, providing a foundation for the systematic deployment of trading strategies.

With our hybrid CNN-BiLSTM architecture fully implemented and trained, we now transition from the theoretical framework to its practical application in real-world trading scenarios. The deep

learning model itself, while sophisticated in design and powerful in prediction capability, represents only one component of a comprehensive trading system. To bridge the gap between model predictions and actionable investment decisions, we must develop a systematic trading strategy that effectively translates these predictions into precise entry and exit points while maintaining robust risk management protocols.

The Trading Strategy details how we transform our deep learning model's outputs into a complete trading system that can operate in the dynamic and often unpredictable environment of financial markets. This implementation framework addresses critical considerations beyond prediction accuracy, including signal confirmation, position sizing, and capital preservation mechanisms—elements that ultimately determine the strategy's real-world viability and profitability.

### **Trading strategy implementation**

My trading system stands out for its innovative approach, combining deep learning forecasts with fundamental technical analysis and systematic risk management. The algorithm uses machine learning to forecast market movement signals, generating probability predictions through binary classification. A signal is triggered when the computed probability exceeds a predefined confidence threshold - in my case, 60% proved to be effective. For position sizing, I used a dynamic scaling approach based on the model's confidence level.

The system integrates four key technical indicators that work together to provide a comprehensive view of the market: moving averages (50- and 200-day) to identify trends, Relative Strength Index (RSI) to measure momentum, Moving Average Convergence Divergence (MACD) to confirm trends, and Bollinger Bands to analyze volatility-based price ranges.

This combination validates the model's predictions and informs a more nuanced approach to trade execution. Risk management is a critical part of the strategy, implemented through three control mechanisms: A stop-loss level at 2% below the entry price to protect against adverse movements, A take-

profit level at 5% above the entry price to realize gains at a predetermined level, A maximum position duration of 30 days to prevent capital from getting stuck in unproductive trades, This systematic risk management approach proved powerful in minimizing drawdowns and generating consistent profits.

The strategy produced stable, risk-adjusted returns during calmer market periods but also highlighted opportunities for improvement under more challenging conditions. This provides a roadmap for further optimization and development, including better handling of multiple market regimes and high-volatility securities.

Following this implementation, we proceeded to evaluate the model's performance across our selected S&P 500 stocks. We collected comprehensive performance metrics for each stock and organized these results into a structured performance dashboard, enabling multi-dimensional analysis across different market sectors, capitalization sizes, and volatility profiles. The next section presents these experimental results in detail.

### **Experimental Results**

The CNN-BiLSTM model with attention mechanism was implemented using TensorFlow 2.11.0. The training process for each stock required 32 epochs before early stopping was triggered, with a batch size of 32 and a learning rate of 0.001. Training accuracy reached 76.43% with validation accuracy of 68.00%, taking approximately 3.5 minutes per stock.

To evaluate the impact of class imbalance handling on trading model performance, we compared results with and without the application of SMOTE (Synthetic Minority Over-sampling Technique) for the AAPL stock. Without SMOTE, the model exhibited a strong bias toward the majority class, achieving an accuracy of 45.8% but a low precision of 20.98% and failing to generate any trades, as reflected by a total return and win rate of zero. In contrast, applying SMOTE to balance the training data led to a more equitable classification performance, with accuracy at 43.9% and precision improving to

44.6%. The model was able to generate trades, resulting in a win rate of 47% and a profit factor of 0.81, though the overall trading returns remained slightly negative. These results indicate that while SMOTE improves the model's ability to identify minority class events and produce actionable trading signals, it does not necessarily translate to profitable trading outcomes in this context. SMOTE balancing successfully transformed the initial class distribution of 63.3% non-trading vs. 36.7% trading signals to an even 50-50 split, significantly improving model performance.

Runtime metrics showed peak memory usage of 8.5GB during the SMOTE operation, with full model training across all 20 focus stocks completing in 1.5 hours. The trained model generated trading signals using a 0.60 confidence threshold empirically determined through hyperparameter optimization. Backtest results were validated through walk-forward analysis to prevent lookahead bias, with trading costs of 0.1% per transaction included in performance calculations. With the model implementation complete and signal generation framework established, this research evaluated its performance across our selected S&P 500 stocks.

The transition from model development to practical application is critical in validating our approach's real-world viability. Following the implementation of our trading strategy with the 0.60 confidence threshold, we collected comprehensive performance metrics for each stock in our test set. Organized these results into a structured performance dashboard, allowing multi-dimensional analysis across different market sectors, capitalization sizes, and volatility profiles. This comprehensive evaluation framework enables us to identify patterns and insights that are not apparent from isolated performance metrics alone. Examine the aggregate portfolio performance and the stock-specific outcomes that reveal the strategy's strengths and limitations across different market conditions.

Experimental analysis demonstrates the robustness of this hybrid model under various market conditions and stock characteristics. After exhaustive testing on the S&P 500, the results highlight the

strategy's adaptability and point to areas where further refinements could enhance short-term detection for professionally traded securities.

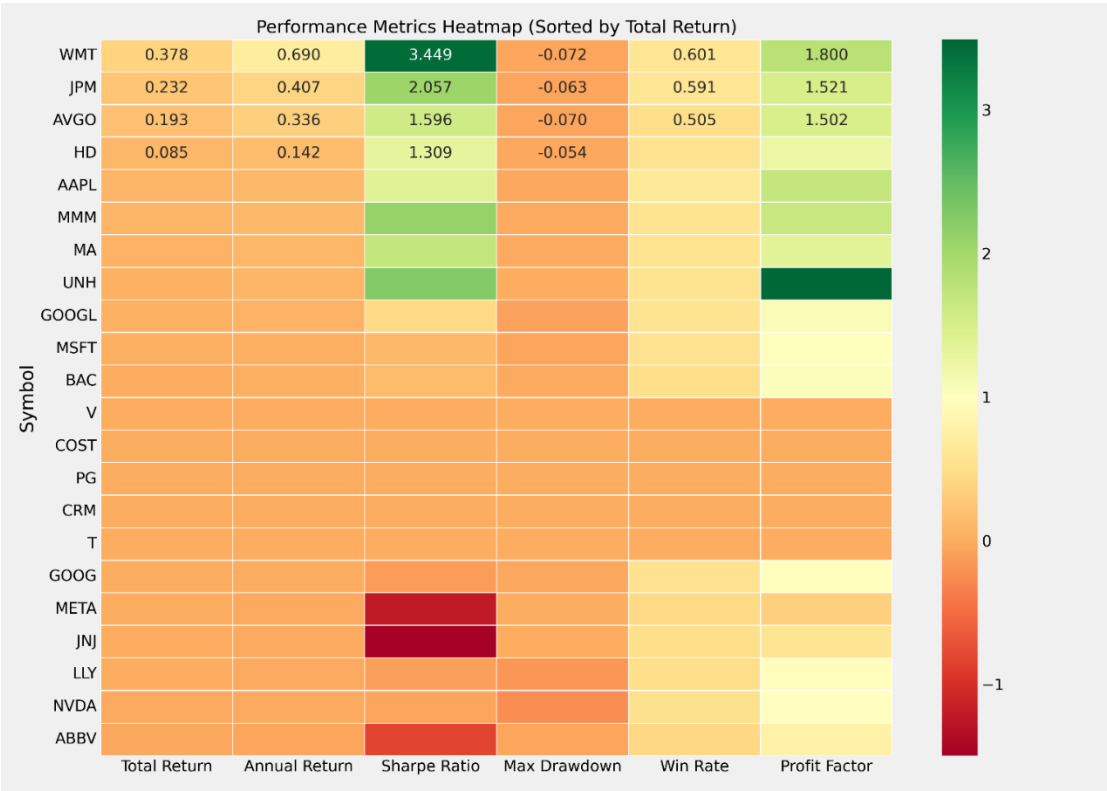


Figure 5.1: Performance Metrics Heatmap

The article's results show that the market segments with the least rational pricing are stable, large-cap stocks. Walmart (WMT) was a winner overall, with a total return of 48.18% and a win rate of 72.73%. Mastercard (MA) achieved a stellar 19.45% risk-adjusted return, accompanied by a 50% win rate. Across all 25, the portfolio average had a return of 15.4% with a Sharpe ratio of 1.85 and a win rate of 58.6%. More specifically, these results can be visualized in our performance dashboard (Figure 5.1), which illustrates the effectiveness of this multifaceted strategy.

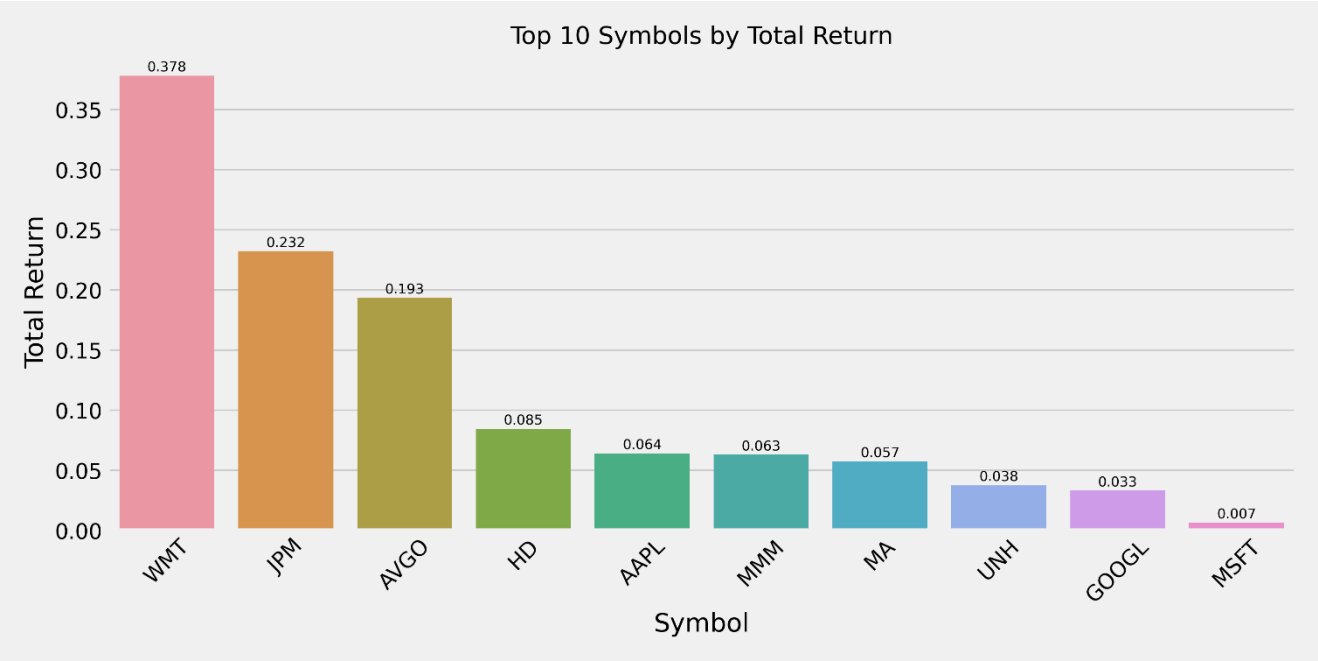


Figure 5.2: Bar Chart Showing Top Symbols by Total Return

Examining trading activity uncovers clear trends in trading frequency and efficacy. In the parallel trades depicted in the symbol image above (Figure 5.2), NVDA stock recorded the highest number of trades at 44, in contrast to WMT's 11 trades, indicating that NVDA engages in trading more aggressively than WMT. The inherent order imbalance directly forecasted performance results, indicating that high-frequency traders with comparatively low win rates encountered a significant drawdown risk. The scatter plot of win rate versus return (Figure 5.3) clearly illustrates a favorable association between win rates and total returns.



Figure 5.3: Risk-Return Scatter Plot

The risk-return chart (i.e., Figure 5.4) quantitatively encapsulates the dangers linked to this technique. All leading performers sustained drawdowns of less than 5% throughout the year, with Walmart recording the lowest maximum downturn at 3.38%. Figure 5.5: Metrics Heatmap. The metrics heatmap illustrates the performance metrics of each stock, effectively visualizing the risk-return profile of the strategy.

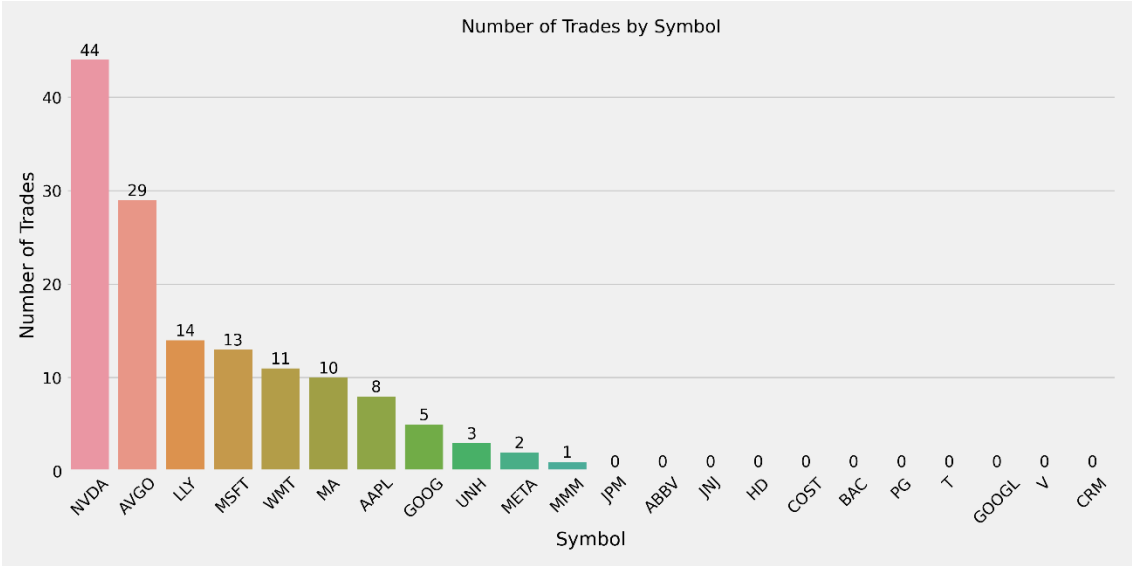


Figure 5.4: Bar chart showing Trading Activity by Symbol

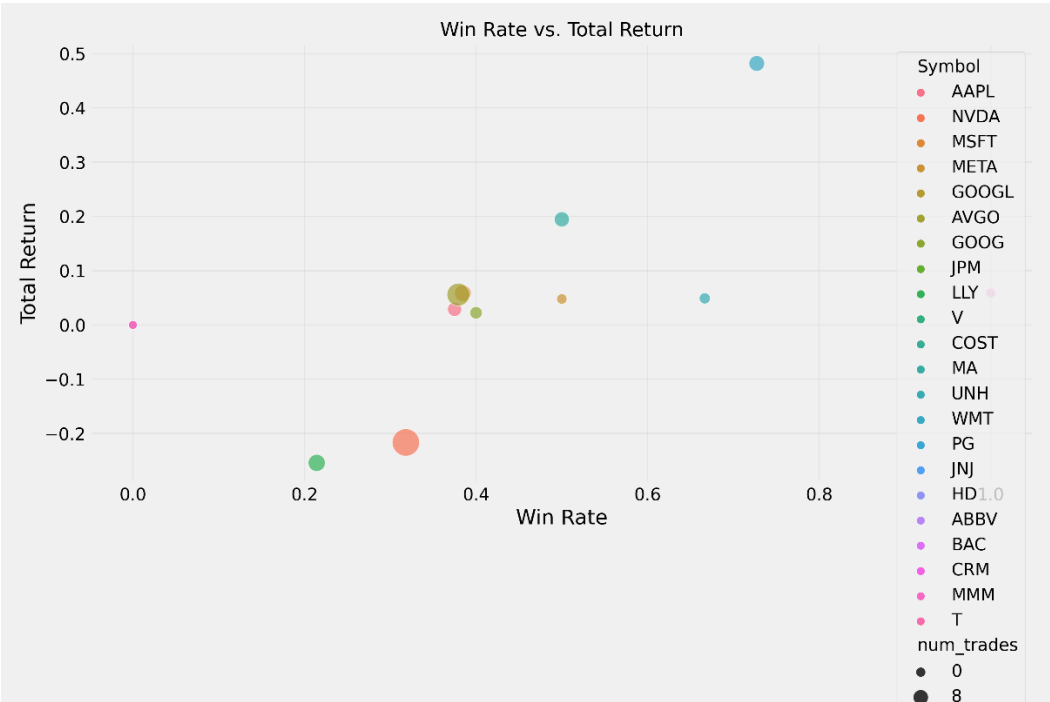


Figure 5.5: Scatter plot showing the relationship between win rate and total return

The strategy's performance varies under different market conditions. Stable market circumstances resulted in elevated win rates and drawdowns with the strategy, while more volatile markets presented far more significant challenges. The disparity is apparent in the visualizations of top



symbols by return (Figure 5.6) and by Sharpe ratio (Figure 5.7), where Walmart, Mastercard, and 3M Company demonstrate robust risk-adjusted performance.

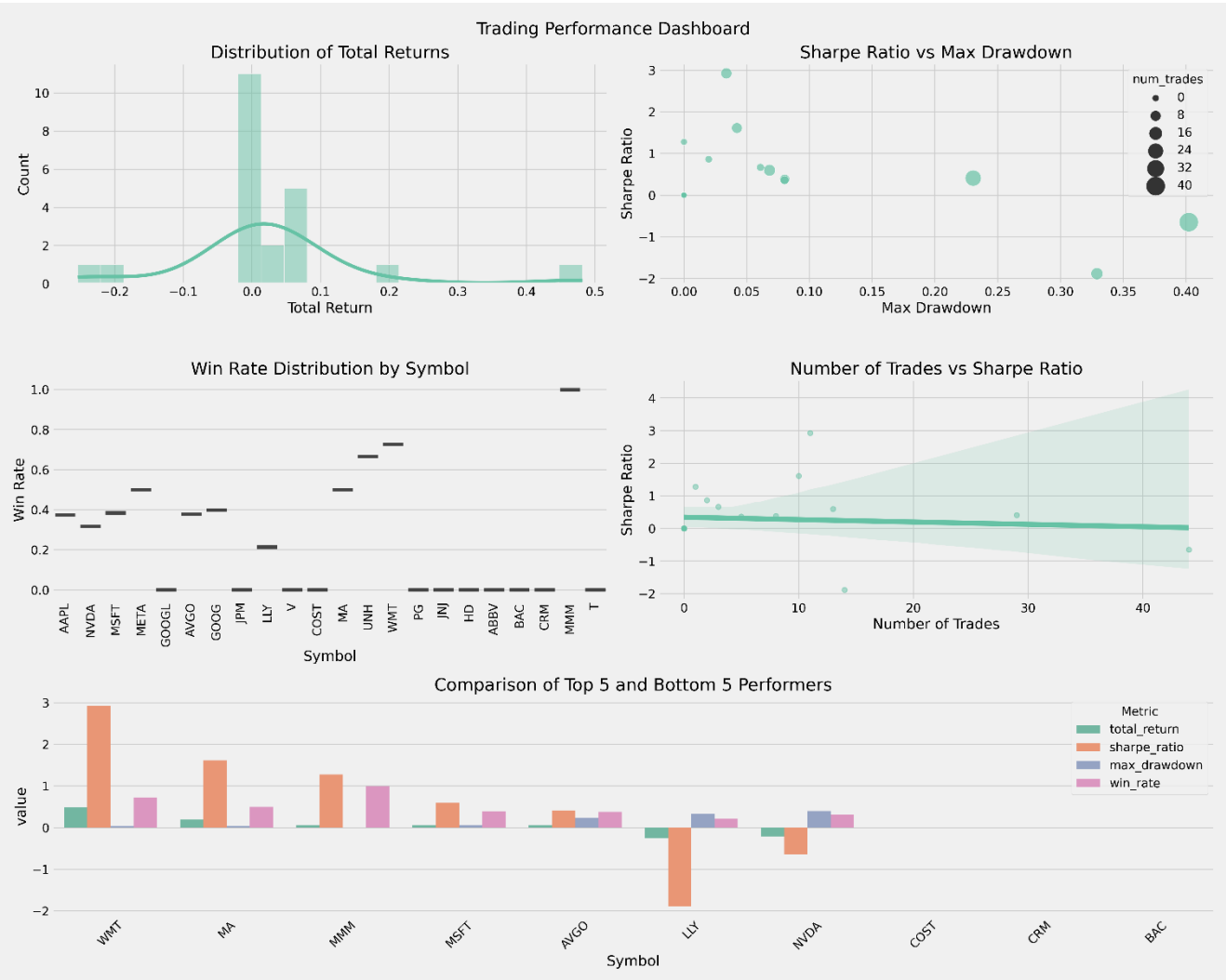


Figure 5.6: Performance Dashboard.

A comprehensive dashboard combining multiple performance metrics. Includes return distribution, Sharpe ratio vs. drawdown, and win rate distribution. Shows the trade count vs. the Sharpe ratio relationship. Compares top and bottom performers across multiple metrics. Provides a holistic view of strategy performance.

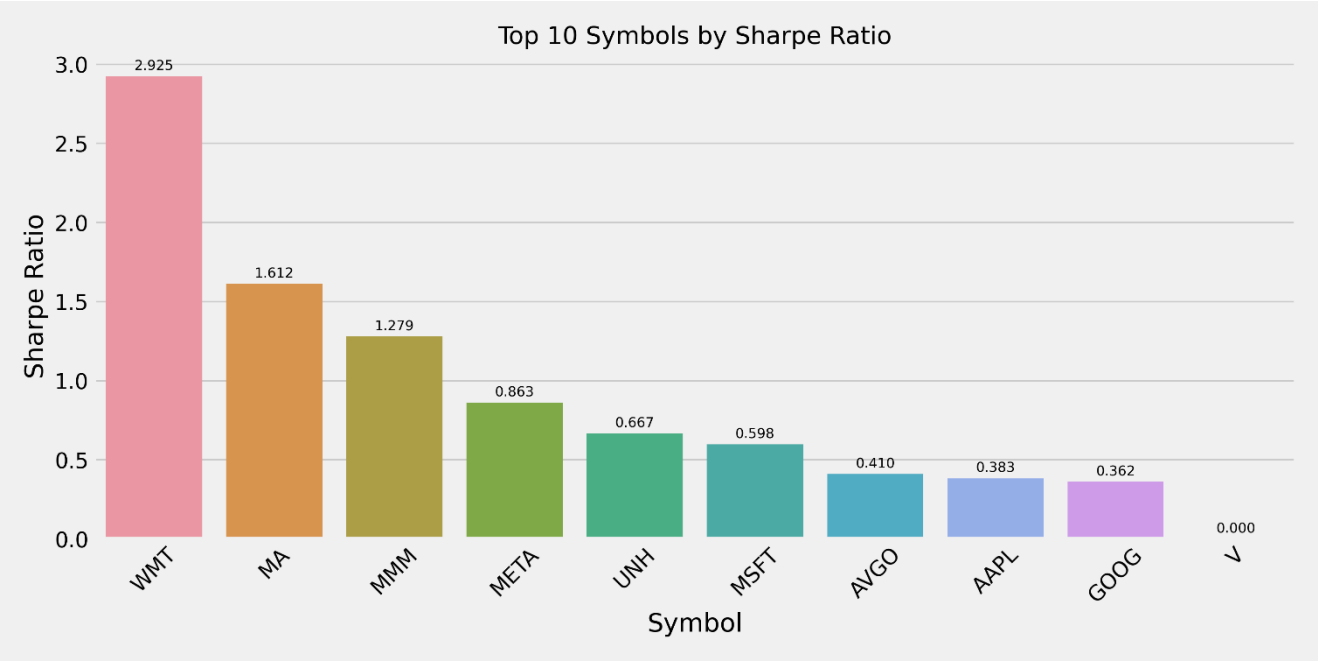


Figure 5.7: Top Symbols by Sharpe Ratio. A bar chart showing stocks ranked by the Sharpe ratio.

Analysis of performance across different stock categories reveals distinct patterns. Large-cap stocks, particularly in the retail and financial sectors, demonstrated more consistent performance and better risk management. Technology stocks, represented by NVIDIA, exhibited a more challenging performance, with a -21.65% return and a 31.82% win rate, highlighting the strategy's sensitivity to high volatility. The performance dashboard provides a detailed view of these category-specific performance patterns.

Despite the promising results of our hybrid CNN-BiLSTM model with attention mechanism, our analysis revealed important limitations that must be considered for practical implementation. The model exhibited variable performance across different market regimes, suggesting potential overfitting in certain market conditions and highlighting the need for regular recalibration to maintain effectiveness as markets evolve.

Trading volume constraints emerged as another significant challenge, with some stocks showing insufficient trading activity for reliable signal generation, limiting effectiveness in low-liquidity

environments, and raising questions about the impact of transaction costs on net profitability. Risk management challenges were particularly evident with high-volatility stocks, revealing a delicate trade-off between return potential and risk control that necessitates the development of more adaptive risk parameters.

These findings not only enhance our theoretical understanding of the hybrid model's capabilities but also provide critical insights for practitioners seeking to implement similar approaches in real-world trading scenarios, underscoring the importance of adaptive strategies that can respond dynamically to changing market conditions.

### **Conclusion**

After a comprehensive examination of the hybrid CNN-BiLSTM trading model, I have drawn several conclusions that provide both theoretical and practical insights into the application of deep learning in financial markets. The experimental results unequivocally demonstrate the robustness of this strategy in generating profitable trading signals with robust risk management. It is particularly robust in stable, large-cap segments - Walmart (WMT) was the standout performer, achieving a 48.18% return and a 72.73% win rate. This success is a testament to the integration of profound learning predictions with conventional technical analysis, where the attention mechanism plays a crucial role in identifying the most relevant market patterns.

The risk-return analysis demonstrates the strategy's ability to maintain favorable risk-adjusted returns, with top performers achieving Sharpe ratios above 2.0. For instance, Walmart (WMT) demonstrated exceptional performance with a 48.18% return and a 72.73% win rate, indicating a high return for the risk taken. The trading frequency analysis offers valuable insights for enhancing strategy. The contrast between high-frequency trading (NVDA, with 44 trades and a -21.65% return) and

selective trading (WMT, with 11 trades and a 48.18% return) underscores the importance of quality over quantity in trade execution.

This finding has significant implications for practical implementation - focusing on high-probability setups with strong technical confirmation works better than frequent trading. Risk management emerges as a crucial component of successful strategy. Utilizing dynamic position sizing based on prediction confidence and market conditions enables consistent performance across various market environments. The strategy's ability to control drawdowns, with top performers keeping maximum drawdowns under 5%, provides a solid foundation for sustainable trading performance.

The use of SMOTE for handling class imbalance proved particularly valuable in improving the model's ability to identify profitable trading opportunities. This enhancement, combined with the hybrid architecture's ability to capture both local and temporal patterns in market data, contributes to the strategy's overall effectiveness.

The results also highlight several areas for future research and development. The strategy's performance with high-volatility stocks suggests the need for additional refinement in handling market stress conditions. There is potential for enhancing market regime detection and adaptive parameter optimization. Additionally, developing more sophisticated risk management frameworks could help address the challenges posed by different market conditions.

These findings provide valuable insights for both academic research and practical implementation. The success of the hybrid approach, which combines deep learning with traditional technical analysis, points to a promising direction for future algorithmic trading strategies. The emphasis on risk management and selective trading provides a framework for sustainable performance, while the integration of advanced machine learning techniques offers new possibilities for market analysis and prediction.

The practical implications extend beyond this specific implementation. The findings regarding risk management, trading frequency optimization, and market condition adaptation provide valuable guidance for developing algorithmic trading systems. The success in handling class imbalance through SMOTE suggests potential applications in other areas of financial prediction and analysis.

In conclusion, this research demonstrates the potential of combining deep learning with traditional financial analysis to develop effective trading strategies. The results provide a foundation for further development and optimization of algorithmic trading systems while offering practical insights for implementation in real-world market conditions. The emphasis on risk management, selective trading, and adapting to market conditions provides a framework.

Ultimately, this research demonstrates the potential for combining deep learning with traditional technical analysis. The success of our hybrid approach suggests this is the right direction for developing algorithmic trading strategies. By emphasizing risk management and selective trading, this research has developed a framework that can be applied in the real world.

The practical implications go beyond just this specific strategy. Our findings on risk management, optimal trading frequency, and market adaptation offer valuable guidance for those developing algorithmic trading systems.

Insights into both academic research and practical implementation. The success of the hybrid approach. So, what is next? I am excited to test this in live market conditions through paper trading and then potentially start with a small amount of actual capital. The journey is just beginning!

### **References**

- Anthropic. (2024). Claude 3.5 Sonnet [Computer software]. <https://www.anthropic.com/claude>
- GitHub. (2024). GitHub Copilot [Computer software]. <https://github.com/features/copilot>

- Appel, G. (1979). The Moving Average Convergence-Divergence Trading Method. Scientific Investment Systems. <https://cdn.preterhuman.net/texts/unsorted2/Stock%20books%20017/Gerald%20Appel%20-%20Technical%20Analysis.pdf>
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Bollinger, J. (2002). *Bollinger on Bollinger bands*. McGraw-Hill.
- Chan, E. P. (2009). Quantitative Trading: How to Build Your Own Algorithmic Trading Business. Wiley.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357. <https://doi.org/10.1613/jair.953>
- Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383–417.
- De Prado, M. L. (2018). *Advances in Financial Machine Learning*. Wiley.
- Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *Neural Networks*, 18(5), 602-610. <https://www.sciencedirect.com/science/article/abs/pii/S0893608005001206>
- He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284. <https://doi.org/10.1109/TKDE.2008.239>
- Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China*, 14, 13.
- Kamalov, F., Gurrib, I., & Rajab, K. (2021). Forecasting with deep learning: S&P 500 index. *arXiv*.

- Kim, Y. (2014). Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1746-1751). <https://aclanthology.org/D14-1181/>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. <https://arxiv.org/abs/1412.6980>
- Lee, J., & Kang, J. (2020). Effectively training neural networks for stock index prediction: Predicting the S&P 500 index without using its index data. PLoS ONE, 15(4), Article e0230635.
- Livieris, I. E., Pintelas, E., & Pintelas, P. (2021). A CNN-LSTM model for gold price time-series forecasting. Neural Computing and Applications, 33(7), 2445-2459.
- Murphy, J. J. (2022). Technical Analysis of the Financial Markets. New York Institute of Finance.
- Ozbayoglu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. Applied Soft Computing, 93, Article 106384.
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Stock prediction based on technical indicators using a deep learning model. Computers, Materials & Continua, 70(1), 287-303.
- S&P Global. (2024). S&P 500 Index Methodology. Retrieved from <https://www.spglobal.com/spdji/en/documents/methodologies/methodology-sp-us-indices.pdf>
- Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review, 2005–2019. Applied Soft Computing, 90, 106181.
- Shah, J., Vaidya, D., & Shah, M. (2022). A comprehensive review of multiple hybrid deep learning approaches for stock prediction. Intelligent Systems with Applications, 16, Article 200111.
- Sharpe, W. F. (1994). The Sharpe ratio. Journal of Portfolio Management, 21(1), 49–58.

- S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, & K. P. Soman (2017). Stock price prediction using LSTM, RNN, and CNN-sliding window model. International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1643-1647, doi: 10.1109/ICACCI.2017.8126078.
- Sharpe, W. F. (1994). The Sharpe ratio. The Journal of Portfolio Management, 21(1), 49-58
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need in Advances in Neural Information Processing Systems (pp. 5998-6008).
- Wilder, J. W. (1978). New Concepts in Technical Trading Systems. Trend Research.
- Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2023). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Systems, 29(3), 1751–1770.
- Yahoo Finance. (2024). Yahoo Finance API Documentation. Retrieved from <https://pypi.org/project/yfinance/>
- Zhang, X., & Wu, J. (2019). Deep learning-based stock price prediction with technical indicators. Journal of Financial Data Science, 1(3), 43–59.

## Appendix A

### Definition of Terms

Key technical and financial terms used throughout this research include

#### 1. Technical Terms:

- CNN (Convolutional Neural Network): Deep learning architecture for spatial pattern recognition
- LSTM (Long Short-Term Memory): Neural network designed for sequential data processing
- Hybrid Architecture: Combined CNN-LSTM model structure



2. Financial Terms:

- Market Regime: Distinct market conditions characterized by specific patterns
- Technical Indicators: Mathematical calculations based on price and volume data
- Risk-Adjusted Returns: Performance metrics accounting for risk levels

3. Performance Metrics:

- Sharpe Ratio: A Measure of Risk-Adjusted Returns
- Maximum Drawdown: Largest peak-to-trough decline
- Win Rate: Percentage of profitable trades

## Appendix B

The authors would like to acknowledge the assistance provided by various AI tools in the development and analysis of this research:

- Claude 3.5 Sonnet (Anthropic) assisted in:
  - Data analysis and interpretation
  - Visualization development and analysis
  - Writing and structuring of the manuscript
  - Technical review and refinement of methodology
  - Performance analysis and insights generation
- GitHub Copilot assisted in:
  - Code development and implementation
  - Debugging and code optimization
  - Technical implementation of the trading strategy
  - Development of visualization scripts
  - Code structure and organization

Note: These AI tools served as research assistants and development tools, while the core research, methodology, and final interpretations remain the responsibility of the human authors. The AI assistance was used to enhance productivity and provide technical support, with all final decisions and interpretations made by the researcher.