

End-to-End Optimization of High-Frequency ETF Trading with BiLSTM and FinBERT-Driven Sentiment Analysis

Zihan Su^{1*}, Yancong Deng²

¹School of Mathematical Sciences, University of Nottingham, Nottingham, UK

²Jacobs School of Engineering, University of California San Diego, San Diego, CA, USA

Email: *smyzs4@nottingham.ac.uk

How to cite this paper: Su, Z. H., & Deng, Y. C. (2024). End-to-End Optimization of High-Frequency ETF Trading with BiLSTM and FinBERT-Driven Sentiment Analysis. *Modern Economy*, 15, 1026-1042. <https://doi.org/10.4236/me.2024.1510053>

Received: July 17, 2024

Accepted: October 26, 2024

Published: October 29, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents a comprehensive approach to developing effective trading strategies for Exchange-Traded Funds (ETFs) by leveraging Long Short-Term Memory (LSTM) networks and sentiment analysis. LSTM networks are well-suited for modeling sequential data and capturing temporal dependencies, making them ideal for predicting stock market trends based on historical data. To further enhance prediction accuracy, sentiment analysis is integrated into the model, allowing for the evaluation of market sentiment derived from financial news and social media. This paper employs FinBERT, a specialized financial language model, pre-trained with masked language modeling (MLM) and next sentence prediction (NSP) tasks, to analyze financial text data. The study examines the effectiveness of various LSTM and Bidirectional LSTM (BiLSTM) architectures in predicting ETF price movements and evaluates their performance using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Additionally, two trading strategies, ultra-high frequency and high frequency, are implemented to assess the practical applicability of the proposed models. The results demonstrate that the combination of LSTM networks and sentiment analysis provides more accurate predictions and leads to profitable trading strategies. The findings suggest that this integrated approach offers a robust framework for developing advanced trading systems in financial markets.

Keywords

LSTM, Trading Strategy, Sentiment Analysis, ETF, Stock Prediction

1. Introduction

The financial sector has made notable strides in applying machine learning

techniques to improve trading strategies. Among the various types of recurrent neural networks (RNNs), long short-term memory (LSTM) networks have attracted significant attention for their capability to model sequential data and capture temporal dependencies. This makes LSTM networks particularly suitable for predicting stock market trends, where historical data plays a crucial role in forecasting future movements.

RNNs are designed to detect patterns in sequential data, such as time series, by preserving a “memory” of previous inputs through hidden states. This feature makes them effective for tasks where the context of earlier data points is essential. However, conventional RNNs struggle with the vanishing gradient problem, which hampers their ability to learn long-term dependencies. LSTMs address this challenge by incorporating a more intricate architecture that uses gating mechanisms to regulate information flow, enabling them to better retain long-term dependencies.

LSTM networks have demonstrated significant potential in financial applications due to their capacity to learn and remember long sequences of historical price data. This makes them ideal for predicting market trends as they can capture complex patterns and temporal relationships within financial time series data. Their structure includes forget gates, input gates, and output gates, which work together to control information flow and ensure that important signals are preserved over time.

Exchange-traded funds (ETFs) are investment funds traded on stock exchanges that represent diversified portfolios of assets and are becoming increasingly popular among investors due to their liquidity and diversification benefits. Developing a robust trading strategy for ETFs is critical to maximizing returns and managing risk in dynamic financial markets. The complexity and volatility of these markets require sophisticated methods to adapt to changing conditions and integrate different data sources.

While LSTM networks are the primary focus of this study due to their effectiveness in handling time-series data, sentiment analysis also plays a supplementary role. Sentiment analysis, a key application of natural language processing (NLP), involves determining the sentiment or emotional tone behind textual data sources such as news articles, social media posts, and financial reports. By gauging public sentiment towards specific stocks or the broader market, traders can make more informed decisions. Although the primary emphasis of this study is on LSTM-based trading strategies, integrating sentiment analysis provides an additional layer of information that can enhance model performance.

The rationale for combining LSTM networks with sentiment analysis lies in the hypothesis that market sentiment, as reflected in news articles, has a significant impact on ETF prices. Integrating sentiment scores with historical price data can provide a more comprehensive view of the factors influencing market movements, potentially leading to more accurate predictions and better trading outcomes.

This study focuses on leveraging LSTM networks to develop and enhance

trading strategies for the ETF market. By incorporating sentiment analysis, we aim to create a more holistic and robust trading strategy that can adapt to the complexities of financial markets. The integration of these advanced techniques addresses the need for sophisticated models capable of incorporating diverse data sources and adapting to market dynamics.

In summary, the use of LSTM networks and sentiment analysis in financial trading strategies offers a promising approach to navigating the complexities of the ETF market. This paper investigates the efficacy of these techniques in predicting ETF prices and formulating profitable trading strategies, with the goal of bridging the gap between advanced machine learning methods and practical applications in financial markets.

2. Related Work

2.1. Development and Impact of LSTM/RNN in Stock Prediction

The use of Long Short-Term Memory (LSTM) networks in financial forecasting has seen significant growth over the past few years. RNNs were originally created to process sequential data by using hidden states that store information from earlier time steps. Nevertheless, conventional RNNs encountered difficulties, particularly the vanishing gradient problem, which limited their capacity to capture long-term dependencies. And LSTM networks is introduced by Hochreiter and Schmidhuber (1997: pp. 1735-1780), addressed these issues by incorporating gating mechanisms that control the flow of information, thus enabling the modeling of long-term dependencies more effectively.

LSTM models have demonstrated considerable success in predicting stock prices and developing trading strategies. For instance, Michańków et al. (2022: pp. 251-264) utilized LSTM networks to predict the values of Bitcoin and the S&P500 index, introducing innovative loss functions to enhance the predictive capabilities of LSTM models in algorithmic trading strategies. Similarly, Murthy et al. (2022) explored the use of LSTM networks for predicting stock price movements of the S&P500, finding that LSTM models performed comparably to baseline linear classification models, highlighting their potential in financial forecasting. Additionally, Kim and Kim (2019) proposed a feature fusion LSTM-CNN model that integrates temporal features with image features from stock chart images, demonstrating superior performance compared to single models.

2.2. Sentiment Analysis and Its Evolution

Sentiment analysis is an important application of natural language processing (NLP) and has evolved to become a crucial tool in financial markets. It involves determining the sentiment or emotional tone behind textual data sources such as social media posts, news and financial reports. The origins of this can be traced back to the early 2000s when researchers started investigating the application of machine learning methods for classifying text according to sentiment. As time passed, advancements in more complex NLP models like BERT and GPT have

greatly improved the precision and dependability of sentiment analysis.

The integration of sentiment analysis with financial models has been shown to enhance trading strategies. [Lin et al. \(2023\)](#) incorporated news sentiment ratios with machine learning models such as LSTM, finding that while sentiment analysis did not significantly improve performance in all cases, it can be a valuable supplement to technical indicators in trading strategies. [Yan et al. \(2023: pp. 1-22\)](#) developed a model that combines Naive Bayes sentiment classifiers with LSTM networks, demonstrating that sentiment indicators significantly enhance forecasting performance compared to models that do not incorporate sentiment data. [Kohsasih et al. \(2022: pp. 1-10\)](#) implemented an RNN-LSTM model for sentiment analysis of financial news, achieving high precision and accuracy, underscoring the model's effectiveness in understanding market trends and informing investment decisions.

2.3. Development of Trading Strategies

The combination of LSTM networks and sentiment analysis for ETF trading strategies represents a relatively novel approach, leveraging both historical price data and market sentiment for enhanced predictions and trading outcomes. The development of trading strategies has traditionally relied on technical and fundamental analysis, but the advent of machine learning has introduced new dimensions to these strategies.

[Chalvatzis and Hristu-Varsakelis \(2019: pp. 123-145\)](#) developed a deep LSTM-based model to predict asset prices and generate profitable trading strategies, emphasizing the importance of integrating predictive models with effective trading strategies. [Wang et al. \(2020: pp. 321-338\)](#) explored the use of LSTM models in conjunction with technical indicators for trading the s300etf, demonstrating that LSTM models can effectively predict trading indices and enhance return on investment. These studies highlight the potential of LSTM networks in developing robust trading strategies that can adapt to the complexities of financial markets.

By integrating LSTM networks with sentiment analysis, researchers aim to develop more comprehensive and adaptive trading strategies. This approach addresses the need for sophisticated models capable of incorporating diverse data sources and adapting to market dynamics, offering promising advancements in ETF trading strategies.

3. Methodology

3.1. Data

As of June 9, 2020, we have collected all data for : DIA (SPDR Dow Jones Industrial Average ETF), EEM (iShares MSCI Emerging Markets ETF), EFA (iShares MSCI EAFE ETF), ERX (Direxion Daily Energy Bull 3X Shares), EWZ (iShares MSCI Brazil ETF), FAS (Direxion Daily Financial Bull 3X Shares), FXE (Invesco CurrencyShares Euro Trust), GDX (VanEck Vectors Gold Miners ETF), HYG (iShares iBoxx \$ High Yield Corporate Bond ETF), IVV (iShares Core S&P 500

ETF), IWF (iShares Russell 1000 Growth ETF), IWM (iShares Russell 2000 ETF), IWO (iShares Russell 2000 Growth ETF), IYR (iShares U. S. Real Estate ETF), JNK (SPDR Bloomberg Barclays High Yield Bond ETF), LQD (iShares iBoxx \$ Investment Grade Corporate Bond ETF), MDY (SPDR S&P MidCap 400 ETF), OIH (VanEck Vectors Oil Services ETF), QLD (ProShares Ultra QQQ ETF), RSX (VanEck Vectors Russia ETF), SDS (ProShares UltraShort S&P 500 ETF), SHY (iShares 1 - 3 Year Treasury Bond ETF), SLV (iShares Silver Trust), SPY (SPDR S&P 500 ETF), SSO (ProShares Ultra S&P 500 ETF), TNA (Direxion Daily Small Cap Bull 3X Shares), TZA (Direxion Daily Small Cap Bear 3X Shares), UPRO (ProShares UltraPro S&P 500 ETF), USO (United States Oil Fund LP), VTI (Vanguard Total Stock Market ETF), VWO (Vanguard FTSE Emerging Markets ETF), VXX (iPath Series B S&P 500 VIX Short-Term Futures ETN), XLB(Materials Select Sector SPDR Fund), XLE(Energy Select Sector SPDR Fund), XLF(Financial Select Sector SPDR Fund), XLI(Industrial Select Sector SPDR Fund), XLU(Utilities Select Sector SPDR Fund), XLV(Health Care Select Sector SPDR Fund), XME(SPDR S&P Metals & Mining ETF), XOP(SPDR S&P Oil & Gas Exploration & Production ETF), XRT(SPDR S&P Retail ETF).

Each ETF contains minute-level and daily-level data respectively. The start date is the date when the ETF starts trading, and the unified end date is June 9, 2020. For details on the data source, see **Table 1**.

Table 1. Data.

Data Type	Time period	Data Source	API/Links
ETFs minute-level transaction data	Terminated on June 8, 2020	Alpha Vantage API	https://www.alphavantage.co
ETFs minute-level transaction data	Terminated on June 8, 2020	Alpha Vantage API	https://www.alphavantage.co

ETFs include minute-level and day-level open, close, high, low, and volume. News articles: daily Twitter comments, grouped by company, using the corresponding news API. Pre-market and after-hours data were filtered out. No missing values were found after inspection.

3.2. LSTM and BiLSTM Model Construction

The Long Short-Term Memory (LSTM) network was developed to overcome the limitations of traditional RNNs, which often struggle with long-term dependencies due to problems like vanishing or exploding gradients when handling lengthy sequences. These issues make it difficult for standard RNNs to effectively learn and retain information over extended time intervals.

LSTMs mitigate this challenge by introducing a more complex internal structure, where each LSTM cell contains specific gates—namely, the input gate, forget gate, and output gate—that regulate the information flow. The input gate controls how much new data enters the cell, the forget gate decides which information from the cell state should be discarded, and the output gate determines how much

of the cell's current state should be passed to the next step.

In an LSTM model, for each time step (t), the input (x_t), hidden state (h_t), and cell state (c_t) are processed to compute the next hidden state and cell state. Given an input sequence $((x_1, x_2, \dots, x_m))$, the LSTM generates corresponding sequences for the hidden states $((h_1, h_2, \dots, h_m))$ and the cell states $((c_1, c_2, \dots, c_m))$. This design enables LSTMs to better handle long-term dependencies and maintain crucial information over time.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot c_t \quad (4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (6)$$

where σ is a logistic sigmoid function, \cdot denotes element-wise multiplication, and C_t is the cell state at time t . $(W_f, W_i, W_c, W_o), (b_f, b_i, b_c, b_o)$ are learnable parameters of the model. The gates f_t, i_t , and (o_t) are the forget gate, input gate, and output gate, respectively. Each LSTM unit has a memory cell with state C_t at time t , controlled by these three gates.

A Bidirectional Long Short-Term Memory (BiLSTM) network is an extension of the unidirectional LSTM that improves the performance of the model by capturing information from both past and future contexts in a sequence. The standard LSTM processes data only in one direction, either forward or backward. However, in many applications, such as natural language processing (NLP), it is beneficial to consider both the previous and future information for a given timestep.

In BiLSTM, two LSTM networks are used. One handles the input sequence from start to end (forward) and the other handles the sequence from end to start (backward). The outputs of these two LSTMs are then connected to form the final output. This bidirectional processing allows the model to understand the sequence more fully.

The BiLSTM model is defined as follows. Let x_t be the input at timestep t , \bar{h}_t be the hidden state of the forward LSTM, and \overleftarrow{h}_t be the hidden state of the backward LSTM. Given a sequence of inputs (x_1, x_2, \dots, x_m) , the BiLSTM computes the forward hidden sequence $(\bar{h}_1, \bar{h}_2, \dots, \bar{h}_m)$ and the backward hidden sequence $(\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_m)$ as follows:

For the forward LSTM:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (7)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (8)$$

$$c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (9)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot c_t \quad (10)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (11)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (12)$$

For the backward LSTM:

$$\tilde{f}_t = \sigma(\tilde{W}_f \cdot [\tilde{h}_{t+1}, x_t] + \tilde{b}_f) \quad (13)$$

$$\tilde{i}_t = \sigma(\tilde{W}_i \cdot [\tilde{h}_{t+1}, x_t] + \tilde{b}_i) \quad (14)$$

$$\tilde{c}_t = \tanh(\tilde{W}_c \cdot [\tilde{h}_{t+1}, x_t] + \tilde{b}_c) \quad (15)$$

$$\tilde{C}_t = \tilde{f}_t \cdot \tilde{C}_{t+1} + \tilde{i}_t \cdot \tilde{c}_t \quad (16)$$

$$\tilde{o}_t = \sigma(\tilde{W}_o \cdot [\tilde{h}_{t+1}, x_t] + \tilde{b}_o) \quad (17)$$

$$\tilde{h}_t = \tilde{o}_t \cdot \tanh(\tilde{C}_t) \quad (18)$$

The final output h_t at each timestep t is the concatenation of the forward and backward hidden states:

$$h_t = [\bar{h}_t; \tilde{h}_t] \quad (19)$$

where:

- σ is the logistic sigmoid function.
- \cdot denotes element-wise multiplication.
- W_f, W_i, W_c, W_o and $\tilde{W}_f, \tilde{W}_i, \tilde{W}_c, \tilde{W}_o$ are the learnable parameters of the forward and backward LSTMs, respectively.
- b_f, b_i, b_c, b_o and $\tilde{b}_f, \tilde{b}_i, \tilde{b}_c, \tilde{b}_o$ are the biases of the forward and backward LSTMs, respectively.

By combining the information from both directions, BiLSTM models can effectively capture the context of each timestep, making them particularly useful for tasks that require understanding the sequence in its entirety, such as text classification, machine translation, and speech recognition.

3.2.1. LSTM Model for Minutes-Level Data

- The input layer's shape depends on whether technical indicators (TIs) are included (**Figure 1**).
- If 'TI = False', the input shape is (time steps, 1), and a single LSTM layer is used.
- If 'TI = True', the input shape is (time steps, number of technical indicators including Bollinger Bands, EMA = 10, 20, 30, RSI, ROX, TSI, sentiment index), and three LSTM layers are used.

Stacking of LSTM layers:

- Each layer contains "50 units" and has return sequences = True, indicating that each LSTM layer returns the full sequence instead of just the last output.

To prevent overfitting, a Dropout layer is added to each LSTM layer with a dropout rate of 0.2.

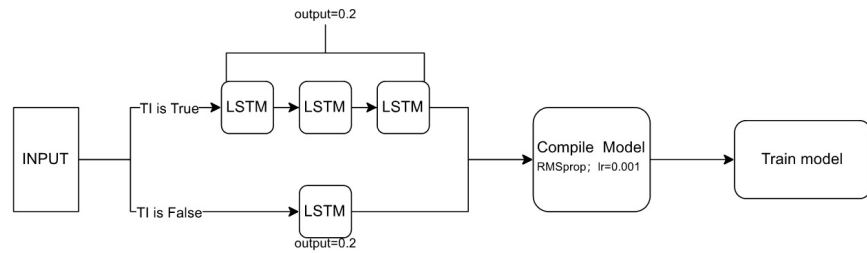


Figure 1. Structure of LSTM model for minutes-level data. (Created by the author)

The specific stacking order is as follows:

- The first LSTM layer returns the full sequence based on the input shape determined by the presence of technical indicators.
- Subsequent LSTM layers each return the full sequence.
- A Dropout layer follows each LSTM layer to reduce the risk of overfitting.
- Finally, a fully connected (Dense) layer outputs a single value (the prediction).

The model is compiled using the RMSprop optimizer, with mean squared error (MSE) as the loss function. During training, the ReduceLROnPlateau callback is used to adjust the learning rate for optimization.

3.2.2. LSTM Model for Day-Level Data

This model is a deep neural network specifically built for handling sequential data, combining both Bidirectional Long Short-Term Memory (BiLSTM) and Long Short-Term Memory (LSTM) layers. The architecture begins by passing the input data through a set of three BiLSTM layers, which are designed to capture dependencies from both previous and upcoming time steps. This makes it especially effective for tasks where context from both directions is crucial.

Following the BiLSTM layers, the model includes two LSTM layers, which process the data in a forward-only direction to retain long-term dependencies in the sequence. At a certain point after the second BiLSTM layer, the model outputs with a probability of 0.4, suggesting some intermediate predictions are made at this stage.

The model is compiled using the Adam optimizer with a learning rate of 0.001, which is effective for optimizing deep learning models. Once compiled, the model is trained to make predictions based on the sequential data it processes. This architecture is well-suited for tasks like time-series forecasting or language modeling, where capturing both short- and long-term dependencies is crucial.

The specific model diagram is shown in **Figure 2**.

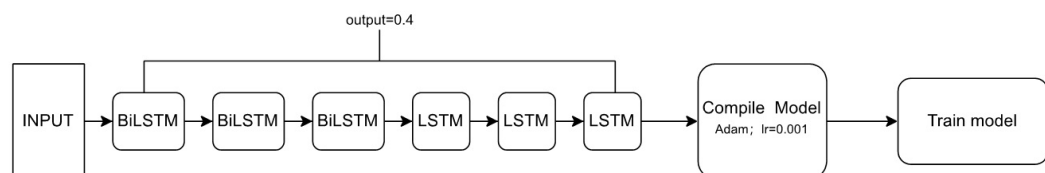


Figure 2. Structure of LSTM model for days-level data. (Created by the author)

3.3. FinBERT Model Construction

FinBERT is a tailored adaptation of the BERT (Bidirectional Encoder Representations from Transformers) model specifically crafted for the financial sector. While BERT has delivered state-of-the-art performance across a wide range of NLP tasks, its training on general-purpose text may not effectively capture the unique characteristics of financial language. To resolve this, FinBERT refines the original BERT model by pre-training it on an extensive financial dataset, including materials such as earnings calls, analyst reports, and financial news articles.

The architecture of FinBERT remains consistent with BERT, utilizing a bidirectional transformer to capture contextual information from both directions in a sentence. This bidirectional approach is crucial for understanding the complex and often jargon-laden language used in financial documents.

3.3.1. Pre-Training Process

FinBERT's pre-training process focuses on two key tasks: masked language modeling (MLM) and next sentence prediction (NSP). In MLM, certain words in a sentence are hidden, and the model is trained to predict these words by leveraging the surrounding context. NSP, on the other hand, tasks the model with determining whether two sentences appear consecutively in the original text, which helps the model grasp the connections between sentences.

The financial corpus used to train FinBERT offers extensive exposure to specialized terminology and industry-specific contexts, greatly improving its ability to comprehend and analyze financial documents.

3.3.2. Fine-Tuning

For our specific application, we fine-tuned FinBERT on financial sentiment analysis, using a labeled dataset of financial documents. This process involved using our labeled dataset to further adjust the model's parameters, optimizing it for our particular use case.

We used specific configurations such as 10 epochs, a batch size of 32, a learning rate of $5e-5$, and the AdamW optimizer with a linear learning rate schedule. This fine-tuning step is essential for adapting the pre-trained FinBERT model to the specific characteristics and requirements of our task, thereby improving its performance on financial sentiment analysis.

By leveraging FinBERT, we aim to achieve more accurate and nuanced analysis of financial texts, benefiting from the model's deep understanding of domain-specific language. In this study, we use FinBERT to process stock comments collected from Twitter. Through NLP techniques, comments are categorized by company and date, and FinBERT provides a sentiment score for each company daily.

3.4. Prediction and Evaluation Metrics

We use the sentiment scores obtained from FinBERT as additional features input into the LSTM model. Using the moving window technique (predicting the 31st data point using the first 30 data points), we can make predictions. Here, we use

the data interval from January 1, 2019, to January 1, 2020, for backtesting.

And we evaluated the results of the six models using RMSE and MAPE.

The Root Mean Square Error (RMSE) is a commonly used metric for evaluating the accuracy of predictive models. It measures the standard deviation of the residuals (prediction errors). The formula for RMSE is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (20)$$

where:

n is the number of samples;

y is the actual value;

\hat{y}_i is the predicted value.

A lower RMSE suggests that the model's predictions are more accurate and align closely with the actual values. Since RMSE involves squaring the errors, it tends to be more responsive to larger deviations, making it effective for identifying substantial prediction errors.

Another commonly used metric for assessing prediction accuracy is the Mean Absolute Percentage Error (MAPE). This metric calculates the average of the absolute percentage differences between the predicted values and the actual values. The MAPE formula is given by:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (21)$$

where: n is the number of samples; y_i is the real value; \hat{y}_i is the predicted value.

MAPE is represented as a percentage, which makes it straightforward to understand. It is particularly valuable for comparing the accuracy of predictions between different models or datasets. However, it can be overly sensitive to very small actual values, which may result in disproportionately large percentage errors.

4. Experiment Result and Discussion

4.1. Trading Strategy

We currently have two trading strategies. One is ultra-high frequency and the other is high frequency. In both transactions, we have introduced the calculation of the tax generated by the transaction. We are citing the tax rate calculation system of Charles Schwab Company. **Table 2** shows the specific transaction fees of Charles Schwab Company.

Table 2. Trading fee.

Fee Type	Fee
Trading commission	\$0 (commission fee)
SEC fee	\$0.22 per \$10,000 of principal
FINRA fee	\$0.000119 per \$1,000 of principal, up to a maximum of \$5.95 per trade

4.1.1. Strategy One

This strategy is completely based on the prediction results in Section 3.4. The strategy is: If it is predicted that the price will rise in $t + 1$ minute, the trading system will decide to buy in t minute. If it is predicted that the price will rise in $t + 2$ minutes, the ETF will be held in $t + 1$ minute. If it is predicted that the price will fall in $t + 1$ minute, the trading system will decide to sell in t minute/no operation. If it is predicted that the price will fall in $t + 2$ minutes, no operation will be performed in $t + 1$ minute. The trading system will traverse the predicted increase of all ETFs in the next minute, select the 4 ETFs with the highest increase, and divide the principal into four parts according to the increase ratio of these four ETFs to buy these four ETFs, respectively.

4.1.2. Strategy Two

This trading strategy targets only one ETF. It is to average the predicted values of $t - t + 5$ minutes, recorded as avg. If the average is greater than the price of t minute and the quotient of the predicted price of $t + 1$ minute and the price of t minute is greater than 1.001, then choose to buy at t minute. If the average is less than the stock price of t minute or the quotient of the predicted price of $t + 1$ minute and the price of t minute is less than 1, then choose to sell/no operation.

4.2. Prediction and Evaluation

Below is a comparison of some prediction results with actual ETF prices.

Figure 3 and **Figure 4** show the prediction results of XLU and EWZ from different models and corresponding real prices.

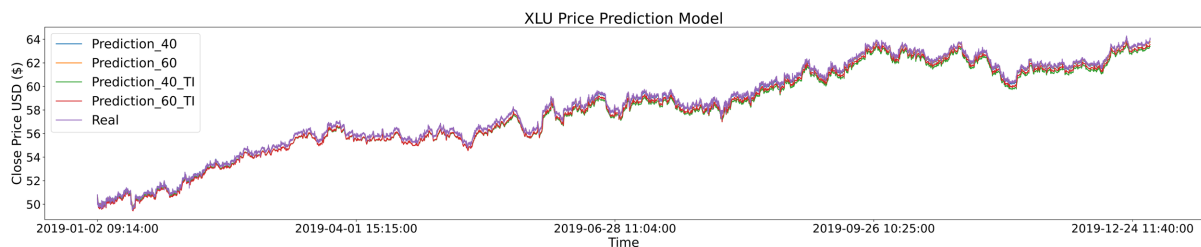


Figure 3. Multiple prediction results and real prices of XLU from different models.

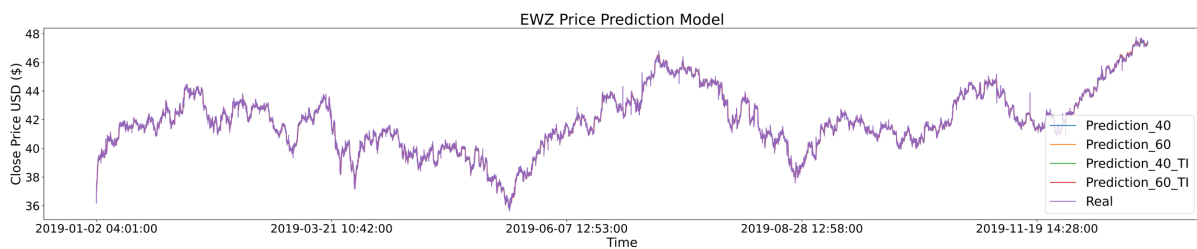


Figure 4. Multiple prediction results and real prices of EWZ from different models.

As shown in **Figure 3**, all models show strong alignment with the actual price movements of XLU, indicating the models' effectiveness in predicting the utility

sector ETF's price. Similarly, **Figure 4** shows the prediction models' performance on EWZ, where the predicted prices closely follow the real stock price, demonstrating the robustness of the models in forecasting the price of this Brazil-focused ETF.

Both figures use the x-axis to represent time (Date) and the y-axis to represent the closing price in USD. The close alignment of the predicted and actual prices in both figures shows that the models perform well across different sectors and geographical markets.

Based on the prediction results, we have calculated the evaluation results of the prediction values of the four models on all ETFs in the dataset. The evaluation results are in **Table 3**.

Table 3. Evaluation of prediction results.

Name	40 TI RMSE	40 TI MAPE	40 RMSE	40 MAPE	60 RMSE	60 MAPE	60 TI RMSE	60 TI MAPE
EFA	0.14	0.00150957	0.09	0.00083525	0.08	0.00078789	0.11	0.00084292
QQQ	0.77	0.00374229	0.47	0.00218225	0.26	0.00095126	0.41	0.0017772
XLP	0.16	0.00229437	0.08	0.00102272	0.13	0.00176219	0.12	0.00156922
IWF	0.88	0.00470314	0.37	0.00182723	0.31	0.00147309	1.36	0.00753543
XLU	0.47	0.00734155	0.13	0.00167056	0.1	0.00121546	0.36	0.00579463
USO	0.35	0.00247923	0.3	0.00212039	0.33	0.00255133	0.48	0.0039887
VTI	0.36	0.00164482	0.27	0.00132594	0.33	0.00178042	0.38	0.00182189
XLV	0.18	0.00138616	0.17	0.00141636	0.15	0.00119946	0.21	0.00163526
TZA	13.93	0.25278375	9.95	0.15911297	9.17	0.14335837	13.77	0.24938059
EWZ	0.15	0.00248604	0.12	0.00189693	0.12	0.00192539	0.14	0.00221278
IYR	0.44	0.00417006	0.15	0.00131632	0.14	0.00123575	0.38	0.00374024
TLT	0.82	0.00520818	0.23	0.00119139	0.22	0.00122811	0.82	0.00556351
DIA	0.67	0.0020243	0.43	0.00119847	0.59	0.00192504	0.86	0.00287819
SPY	0.66	0.00187348	0.61	0.00182387	0.75	0.00236727	0.55	0.00148611
QLD	0.5	0.00410151	0.47	0.00357218	0.35	0.00252546	0.4	0.00292401
OIH	3.89	0.01078901	1.34	0.00345075	1.33	0.00358894	3.91	0.01117578
MDY	1.74	0.00421163	0.86	0.00192961	0.66	0.00128858	2.44	0.00608442
SLV	0.04	0.00134428	0.04	0.00164744	0.04	0.00174469	0.03	0.00120178
MEAN	1.453	0.017449	0.893	0.010530	0.837	0.009606	1.485	0.017311

Test 40 TI means that the time window is 40; that is, the stock price in the first 40 minutes is used to predict the stock price in the following time. And the prediction model of technical indicators such as Bollinger Bands, Relative Strength Index (RSI), and Rate of Change (ROC) is added. Test 40 is a prediction model that uses the stock price in the first 40 minutes to predict the stock price in the following time without adding technical indicators.

From the table, we can see that the average MAPE and RMSE of the model without TI for a time window of 60 are 0.009606 and 0.837, respectively, which

means that this type of model performs best on average. This means that the technical indicators included in TI, such as Bollinger Bands, are not very helpful for prediction. We can also find that all models perform poorly in finding the TZA ETF. We decided to remove this ETF from the ETF pool of the subsequent trading system to ensure the maximum performance of the model.

4.3. Trading Strategy

4.3.1. Strategy One

This trading strategy basically makes a transaction every minute. It is an ultra-high frequency trading strategy. And we run our UHF trading strategy in IYR, TLT, DIA, SPY, QLD, MDY, SLV, OIH, EFA, XLP, IWF, QQQ, USO, VTI, XLV, TZA, EWZ, XLU, ETFs with an initial capital of \$10000. After 100,000 minutes, our profit is 4469.3359\$. The horizontal axis of **Figure 5** and **Figure 6** is minutes, and the vertical axis is money and **Figure 5** shows the process of fund changes.

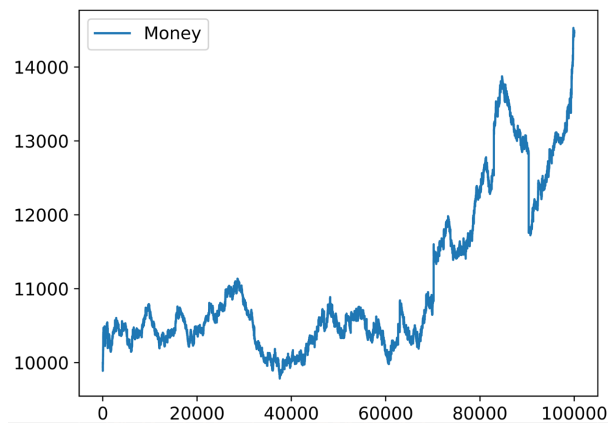


Figure 5. Result of strategy one in minutes-level.

If the transaction is held when the handling fee is greater than the predicted growth value, the result is as shown in **Figure 6**, but these results may need further verification.

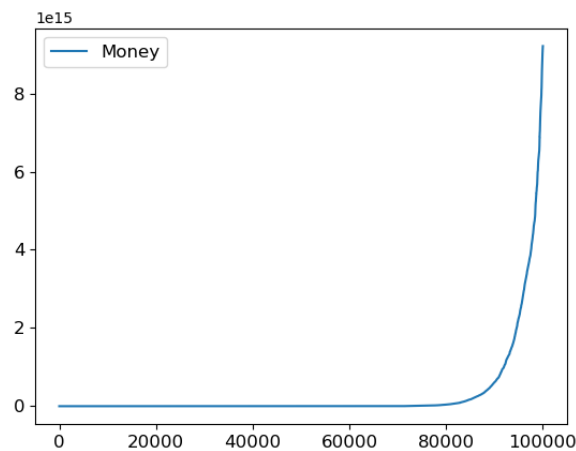


Figure 6. Result of strategy one in minutes-level.

In addition, we replace the data with data in days, that is, record the Open, Close, trading volume, etc. of the ETF every day. And the model structure has been slightly modified. The model does not use the TI indicator. After obtaining the prediction results, this strategy is used for trading simulation, and the following results are shown in **Figure 7**. The vertical axis is money and the horizontal axis is days.

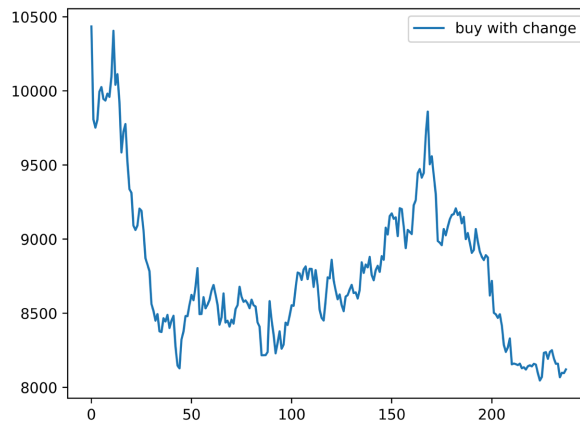


Figure 7. Result of strategy one in days-level.

4.3.2. Strategy Two

By using this high frequency trading strategy into SLV, SPY, EFA and QQQ, the results show that the benefits rate is 199.46%, 150.25%, 149.18% and 121.60%. **Figure 8** shows the specific changes in money. The vertical axis is money and the horizontal axis is the number of transactions.

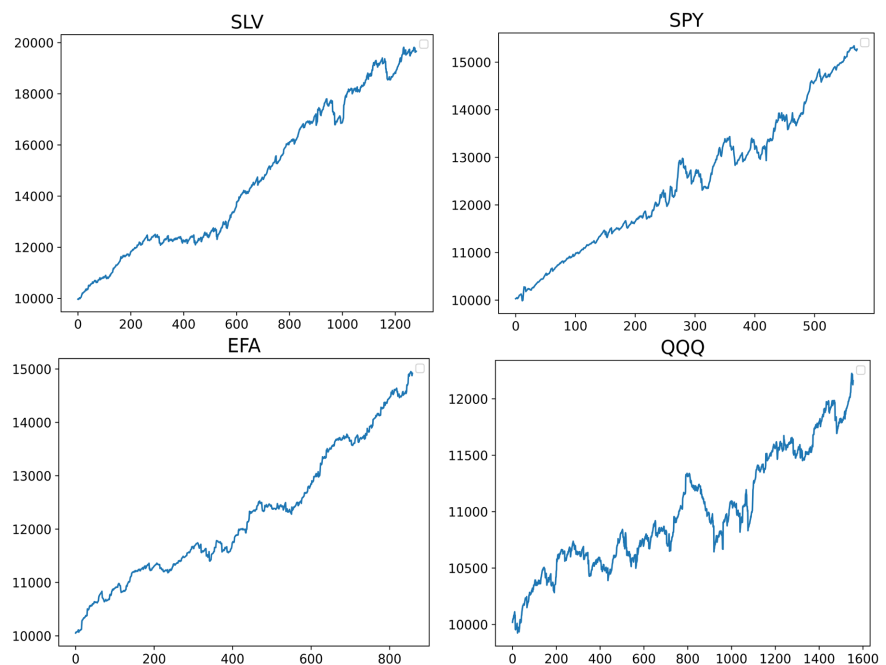


Figure 8. Results of strategy two on SLV, SPY, EFA and QQQ.

4.4. Discussion

When conducting strategy simulation trading, to facilitate comparison, we unified the stock data of the first 100,000 minutes of trading in one year. Strategy 1 carried out a total of 100,000 buying and selling operations. Strategy 2 was performed about 1,000 times. According to calculations, on average, one buying and selling operation takes place every 1.5 hours. This is acceptable for human monitoring.

We compared all ETFs to compare the natural growth of their stocks with the nature of the strategy's effect on the corresponding ETFs. The results are recorded in **Table 4**.

Table 4. Comparison of natural growth rate and growth rate under strategy two.

Name	Natural growth (%)	Growth driven by strategy (%)	60 RMSE
EFA	13.99	49.18	0.08
QQQ	40.41	21.60	0.26
XLP	26.89	30.17	0.13
XLU	25.95	60.10	0.1
USO	35.02	289.28	0.33
VTI	33.08	51.14	0.33
XLV	20.35	21.39	0.15
TZA	-20.85	3.37	9.17
EWZ	24.13	288.60	0.12
IYR	21.14	33.67	0.14
TLT	12.96	34.10	0.22
DIA	24.58	22.99	0.59
SPY	29.49	52.73	0.75
QLD	53.24	349.45	0.35
OIH	48.52	21.09	1.33
MDY	24.25	12.42	0.66
SLV	16.88	96.62	0.04

Among them, the natural growth rate of four ETFs was greater than the growth rate under the influence of strategies. This means that the strategy is working on this ETF to a certain extent.

Through research, it was found that the forecast results of these failed stocks were not very good. Their average "60 RMSE" reached: 0.71. And the average "60 RMSE" of other ETFs was: 0.22. This is a statistically significant difference. Can pass 99% confidence test. (The sample size is 100000, and the confidence interval of 0.22 is (0.2166, 0.2234)). Through this, we can roughly infer whether to launch this strategy based on the performance of the previous model's prediction results on this ETF.

At the same time, TZA, a stock with poor forecast results, has been added to the table above. The purpose is to demonstrate the ability of our strategy to protect the principal when poor prediction results occur to a certain extent.

5. Conclusion

This study presents a comprehensive approach to developing robust trading

strategies for Exchange-Traded Funds (ETFs) by leveraging Long Short-Term Memory (LSTM) networks and sentiment analysis. The application of LSTM networks, known for their ability to model sequential data and capture temporal dependencies, has shown significant promise in predicting stock market trends. By integrating sentiment analysis, which evaluates the emotional tone behind textual data, we enhance the predictive power of our models.

Our research demonstrates that combining LSTM networks with sentiment analysis can provide a more holistic view of the factors influencing market movements. This integrated approach not only improves the accuracy of predictions but also leads to more profitable trading strategies. The evaluation metrics, such as Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), indicate the effectiveness of our models in forecasting ETF prices.

We tested two trading strategies: an ultra-high frequency strategy that makes transactions every minute and a high-frequency strategy targeting specific ETFs. Both strategies yielded substantial profits, highlighting the potential of our approach in real-world trading scenarios. Moreover, our analysis revealed that certain technical indicators, such as Bollinger Bands and Relative Strength Index (RSI), did not significantly enhance prediction performance, suggesting that the LSTM and sentiment analysis combination is sufficiently robust on its own.

In summary, this study bridges the gap between advanced machine learning techniques and practical financial applications. The successful implementation of LSTM networks and sentiment analysis in ETF trading strategies offers a promising direction for future research and development in financial markets. Future work could explore the incorporation of additional data sources and the refinement of trading algorithms to further enhance model performance and trading outcomes.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Chalvatzis, K., & Hristu-Varsakelis, D. (2019). Predicting Asset Prices Using Deep LSTM-Based Models and Technical Indicators. *Journal of Quantitative Finance and Economics*, *5*, 123-145.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*, 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Kim, T., & Kim, H. Y. (2019). Forecasting Stock Prices with a Feature Fusion LSTM-CNN Model Using Different Representations of the Same Data. *PLOS ONE*, *14*, e0212320. <https://doi.org/10.1371/journal.pone.0212320>
- Kohsasih, Y., Hartono, R., & Sembiring, R. (2022). Sentiment Analysis of Financial News Using RNN-LSTM Model. *Journal of Information Systems Engineering and Business Intelligence*, *8*, 1-10.
- Lin, Y., Hu, Y., Wang, M., & Fang, Z. (2023). Incorporating News Sentiment Ratios with Machine Learning Models for Stock Price Prediction. *Finance Research Letters*, *48*, Article 102388.
- Michańkó, P., Oleksy, P., & Stankiewicz, M. (2022). Application of LSTM Networks in

Forecasting the Value of BTC and the S&P500 Index. *Journal of Financial Algorithms and Strategies*, 14, 251-264.

Murthy, A. C., Choudhary, R., & Padhy, S. (2022). Stock Price Movement Prediction Using LSTM Networks. *International Journal of Computational Intelligence and Applications*, 21, Article 2250008.

Wang, Y., Li, Z., & Hu, S. (2020). Application of LSTM Models in Conjunction with Technical Indicators for Trading the s300etf. *Journal of Computational Finance*, 24, 321-338.

Yan, X., Wu, Y., & Zhang, M. (2023). Enhancing Stock Price Prediction with Sentiment Analysis: Combining Naive Bayes Sentiment Classifiers and LSTM Networks. *Journal of Financial Engineering*, 10, 1-22.