WeTrade

Predicting Stock Price Using Artificial Intelligence

Gus Simanson, Max Weinstein, and Drew Zauel

Thomas Jefferson High School for Science and Technology

Dr. Torbert, Dr. Yilmaz

May 26, 2025

Table of Contents

Abstract

WeTrade is an automated stock trading system designed to democratize access to profitable long-term investing by combining technical analysis, news sentiment analysis, and machine learning. Traditional trading tools often require significant financial expertise, time, or costly subscriptions, making them inaccessible to low-income households. Our solution integrates Long Short-Term Memory (LSTM) neural networks trained on historical stock data, technical indicators (MACD, RSI, OBV, SMA), and sentiment scores derived from news articles. During backtesting, the model demonstrated robust predictive accuracy, and in its first week of live trading, it achieved a 4% profit. The accompanying website provides users with a personalized interface to monitor investments, fostering financial literacy. While limitations such as data latency and news bias exist, our system offers a scalable, cost-effective alternative to traditional brokerage services, empowering individuals to participate equitably in the stock market.

Introduction

There is a severe inequality in stock market participation within the United States. Only 34% of households in the bottom half of the U.S. income distribution hold stocks, which collectively account for just 0.6% of the stock market's total value. In contrast, the top 1% of households hold 54% of the market's value (Federal Reserve, 2023). Many working-class families cannot afford brokers, investment advisers, or financial advisors—tools that enable affluent individuals to manage stock portfolios. Additionally, these families lack the time to constantly monitor market trends and determine optimal times to buy or sell.

WeTrade aims to overcome these barriers by automating stock trades through a machine learning model. Traditional trading bots rely on simple algorithms or historical market data alone. Our innovation lies in combining historical data with technical indicators and sentiment analysis of financial news to predict stock performance more accurately. This model is integrated into an accessible website, allowing users to invest effortlessly. By democratizing access to advanced trading tools, WeTrade empowers individuals across demographic backgrounds to build wealth through informed, long-term investments.

Methodology

Data collection

Historical Market Data

We obtained a dataset comprising daily volume, high, low, and closing prices for all NASDAQ, S&P 500, and NYSE-listed companies since 2000 using Yahoo Finance's API. These datasets were organized into folders and updated daily.

Technical Indicators

In addition to the existing historical data, we calculated a set of technical indicators based upon our obtained historical market data. Technical indicators are pattern-based calculations done upon historical stock data which analysts and traders use to make predictions about future movements of that stock price. This practice is called Technical Analysis (Chen). Analysts apply a multitude of technical indicators, but we chose four of the most commonly used.

Our first indicator was Moving Average Converge Divergence, commonly known as MACD, which is an indicator used to determine price momentum in order to identify entry and exit points for prospective buyers and sellers. MACD is calculated by finding the difference between a short and long period Exponential Moving Average (EMA), which is a moving average calculated with a greater weighting on more recent entries. The formula for EMA follows as:

$$EMA_{Today} = \left(Value_{Today} \times \left(\frac{Smoothing}{1+Days}\right)\right) + EMA_{Yesterday} \times \left(1 - \left(\frac{Smoothing}{1+Days}\right)\right)$$

A typical smoothing value used is 2. The MACD line is then constructed by subtracting the 26-Period EMA (26 days) from the 12-Period EMA (12 days).

$$MACD = \text{12-Period EMA} - \text{26-Period EMA}$$

After creating the MACD line, a 9-Period EMA is constructed upon the MACD values, called the signal line. When analyzing a stock, if the MACD falls below the signal line it is a bearish signal, indicating that it may be time to sell. Otherwise, if the MACD rises above the signal line, it signifies a bullish trend, suggesting a possible experience upward momentum. Through looking at these crossovers and divergence, traders are given entry and exit points for their position on the stock. In our dataset, we included the EMA values, the MACD value, the signal value, and the MACD histogram, which is the difference between MACD and the signal.

Fig 1 Image by Sabrina Jiang © Investopedia 2022

The second technical indicator we included was On Balance Volume (OBV), which uses the volume traded (quantity of shares exchanged) to make predictions on the stocks momentum and future price movements (Hayes). The indicator works under the assumption that if the volume of stock traded increases sharply without meaningful change in stock price, the price will eventually greatly increase or decrease (Granville). The formula is:

$$OBV = OBV_{prev} + \begin{cases} \text{volume}, & \text{if close} > \text{close}_{prev} \\ 0, & \text{if close} = \text{close}_{prev} \\ -\text{volume}, & \text{if close} < \text{close}_{prev} \end{cases}$$

OBV = Current on-balance volume level
OBV_{prev} = Previous on-balance volume level
$Close$ = Closing price of day
OBV_{prev} = Closing price of previous day
volume = Latest trading volume amount

OBV is applied to track the patterns of institutional investors, as they tend to buy large quantities of a stock that retail investors are selling without an initial price increase, but eventually that volume of trading drives the price up, which is when institutional investors sell back to retail investors. Therefore, a growing OBV means there is greater buying pressure, leading to a possible price increase, while a decreasing OBV can signify growing selling pressure.

Fig 2. Image by Sabrina Jiang © Investopedia 2021.

The third technical indicator we used was Relative Strength Index, commonly known as RSI, which measures the speed and magnitude of a stocks recent price changes. RSI is an oscillator on a scale from 1 to 100, and it compares a stocks strength on days when prices go up to days when prices go down (Fernando). RSI is split into two calculations; one within the initial 14 periods of available data:

$$RSI_{\text{step one}} = 100 - \left[\frac{100}{1 + \frac{\text{Average gain}}{\text{Average loss}}} \right]$$

And the other that is used outside of the original 14 periods of data:

$$RSI_{\text{step two}} = 100 - \left[\frac{100}{1 + \frac{(\text{Previous Average Gain} \times 13) + \text{Current Gain}}{(\text{Previous Average Loss} \times 13) + \text{Current Loss}}} \right]$$

The average gain/loss used in this calculation is the average percentage gain or loss during the commonly used 14-period lookback timeframe. Periods of price losses are counted as zero in the calculations of average gain, and price increases are counted as zero in the calculations of average loss. An RSI reading greater than or equal to 70 can indicate the stock being overbought by the market, while a 30 or below reading indicates an oversold condition.

Fig 3. Image by Tradingview

Our last technical indicator is Simple Moving Average, also known as SMA. This is the simplest of the indicators, and just takes the average price of the stock over a predetermined period, in our case we chose a 5 day, 10 day, and 20 day SMA. This data is then placed unto a curve, which acts as a more stable and slow moving counterpart.

Fig 4. Image by Sabrina Jiang

Daily Sentiment

We needed to find a way to be able to analyze the current public sentiment on something. Usually, one thing that either reflects that public opinion, or creates the public opinion would be the news. Therefore, we needed to be able to scrape news articles relating to companies and current events. Our first iteration of the scraper was working on scraping articles from: *The Motley Fool, Forbes,* and *Wall Street Journal*.

```
08:53:10.056051 - https://www.forbes.com/sites/greatspeculations/2024/09/05/intel-stock-could
08:53:10.056160 - https://www.forbes.com/sites/greatspeculations/2024/09/05/is-verizons-acqui
08:53:10.056210 - https://www.forbes.com/councils/forbesfinancecouncil/2024/09/05/what-compan
08:53:10.056258 - https://www.forbes.com/sites/greatspeculations/2024/09/05/why-is-xpeng-stoc
08:53:10.056301 - https://www.forbes.com/sites/greatspeculations/2024/09/05/big-tech-buy-sign
08:53:10.056344 - https://www.forbes.com/councils/forbesfinancecouncil/2024/09/05/what-base-r
08:53:10.056403 - https://www.forbes.com/sites/greatspeculations/2024/09/05/pick-boeing-stock
08:53:10.056454 - https://www.forbes.com/sites/andrewrosen/2024/09/05/three-tips-for-successi
08:53:10.056562 - https://www.forbes.com/sites/roystonwild/2024/09/05/vistry-leads-ftse-100-h
08:53:10.056607 - https://www.forbes.com/sites/greatspeculations/2024/09/05/costco-stock-has-
08:53:10.056649 - https://www.forbes.com/sites/roystonwild/2024/09/05/associated-british-food
```

It would run on a loop, checking for new articles every minute, then scraping tickers and analyzing the articles. It would then run the articles through sentiment analysis.

```
MBS- 0.15718 - https://www.fool.com/investing/2024/09
DJT- 0.27500 - https://www.fool.com/investing/2024/09
BRK- 0.50000 - https://www.fool.com/investing/2024/09
BRK- 0.50000 - https://www.fool.com/investing/2024/09
PYPL- 0.55000 - https://www.fool.com/investing/2024/0
DLTR- 0.14286 - https://www.fool.com/investing/2024/0
IJR- -0.25000 - https://www.fool.com/investing/2024/0
RKLB- -0.08330 - https://www.fool.com/investing/2024/
```

The issue we found with this was that there weren't enough articles on these sites to be able to have statistically significant data to work with. The other thing was that we needed to be able to view articles from the past, to train our model on. Therefore, we made the choice to start scraping Yahoo Finance, which had solutions to both of our problems. They had an aggregation of hundreds of articles per day, mostly being about diverse content, which helped solve our number problem. In addition, they had an archive of articles going back to 2012, totalling in 4 million articles for us to scrape to train our model on, so it was a perfect fit for us. We started working on scraping the archive before current news, because then we could train the model in the background while doing other things. To do this, we utilized Yahoo Finance's sitemap:

Yahoo Finance Site Map

Articles

2025

Jan	Feb	Mar	Apr
May			

2024

Jan	Feb	Mar	Apr
May	Jun	Jul	Aug
Sep	Oct	Nov	Dec

These links all lead to months containing links to days, containing links to every article published on that day. Using a python bot with BeautifulSoup4, we were able to scrape every tag iteratively, then create a database of all the possible dates to scrape. Then for every date in that database, we would scrape the list of links that the day contained, and add them to the database.

id	day	month_id	url	processed_percent	added_urls
Filter	Fi...	Filter	Filter	Filter	Filter
1	1	1	https://finance.yahoo.com/sitemap/2025_01_01	0.0	1
2	2	1	https://finance.yahoo.com/sitemap/2025_01_02	0.0	1
3	3	1	https://finance.yahoo.com/sitemap/2025_01_03	0.0	1
4	4	1	https://finance.yahoo.com/sitemap/2025_01_04	0.0	1
5	5	1	https://finance.yahoo.com/sitemap/2025_01_05	0.0	1
6	6	1	https://finance.yahoo.com/sitemap/2025_01_06	0.0	1
7	7	1	https://finance.yahoo.com/sitemap/2025_01_07	0.0	1
8	8	1	https://finance.yahoo.com/sitemap/2025_01_08	0.0	1

So now we had the link to every article since 2012, totalling to around 4.1 million articles that we needed to scrape. We tried building something that would just grab articles as fast as possible, analyze them, and repeat, but we ran into several problems. The largest was that Yahoo Finance seemed to limit us after around 400 articles in a minute. This meant that it would take around 40 days of optimal scraping to scrape every possible article. That optimal scraping also didn't seem to be possible, with Yahoo changing their site pretty often, which would disturb the bot.

Therefore, to increase the speed of the scraping, we needed a way to bypass the rate limit that Yahoo was setting on us. Our first thought was using multiple servers, which worked, but it wasn't efficient enough. We then thought of using proxies, specifically IPv4 proxies. The way those worked is that our network requests could be rerouted through a computer in the middle, and Yahoo would see the traffic coming from somewhere else. With enough proxies,Yahoo wouldn't be able to tell it was all from the same person, and the requests would go through the ratelimit. The issue with those proxies was cost. We could buy a dedicated proxy for around $1.5 a month, with each of those getting us 400 requests / minute. We could also buy unlimited proxies and pay per network usage, but our total usage would equate to around $1200. Therefore, they didn't seem to be viable.

However, we did remember that websites also take requests from IPv6 connections. Because it's a newer protocol, and there are so many more possibilities when creating an IPv6 address, they were much cheaper. In fact, we were able to get a whole /48 subnet of addresses (2^80 addresses) for completely free. The way we did this was through hurricane electric, which offered a tunnel broker for computers that were unable to upgrade to IPv6 protocol. They assigned us a range, and our computer would be able to send the network through any address in that range. Now the only issue was finding a server that would work, because most of the major ones had IPv6 and were blocked from accessing the tunnel broker. After going through 9 companies, we finally found a provider (RamNode) that was cleared by Hurricane Electric.

When writing the script for the scraping, we needed to be able to assign to the addresses and send data through them as fast as possible, therefore, we were writing code at the socket level. To assign to the address we wanted to use, because it was non-local, we used a trick called ip_freebind, which allowed the server to bind the socket to an address not in its local range. When doing that, the request was then able to be routed through the address specified. After that, we were able to send around 50,000 - 60,000 requests per minute to Yahoo, being bottlenecked by the server bandwidth. However, that was more than enough. Because of storage issues, we would scrape an article, and analyze it in-memory, so we only had to store the sentiment value and not the raw data from the article. We actually had to scale back the amount of requests being sent because of the cpu being overpowered by the sentiment being run. Overall, the solution worked well and we were able to build a database of the dates, articles, tickers, and respective sentiment values.

id	url	title	date
Fi...	Filter	Filter	Filter
1	https://finance.yahoo.com/news/wall-stree...	Wall Street's Ambitions in China Run Into a...	2024-01-04 00:00:11.000000
2	https://finance.yahoo.com/news/quiet-...	Quiet Income Kings: 3 Under-the-Radar ...	2024-01-01 00:07:00.000000
3	https://finance.yahoo.com/news/returns-...	Returns On Capital Are Showing Encouragi...	2024-01-01 00:09:16.000000
4	https://finance.yahoo.com/news/diversified...	Diversified United Investment Insider Ups ...	2024-01-01 00:16:35.000000
5	https://finance.yahoo.com/news/nanalysis-...	Nanalysis Scientific Corp. Announces Gran...	2024-01-04 00:11:00.000000
6	https://finance.yahoo.com/news/pine-trail-...	Pine Trail REIT Announces Move to Quarter...	2024-01-04 00:12:00.000000
7	https://finance.yahoo.com/news/estimatin...	Estimating The Intrinsic Value Of ALS ...	2024-01-01 00:38:37.000000
8	https://finance.yahoo.com/news/why-meta...	Why META Stock May Be the Biggest Tech ...	2024-01-04 00:15:39.000000
9	https://finance.yahoo.com/news/drug-price...	As Drug Prices Rise, $35 Cap on Insulin ...	2024-01-04 00:16:40.000000
10	https://finance.yahoo.com/news/lg-display-...	LG Display Unveils Automotive Display ...	2024-01-01 01:00:00.000000
11	https://finance.yahoo.com/news/heres-why...	Here's Why We're Not Too Worried About ...	2024-01-01 01:00:38.000000

tickers
Filter
UBSG.SW
T,SCCO,AMCR
AWZ.SI
DUI.AX
NSCIF
PINE-UN.V
ALQ,ALQ.AX
META,AMZN
SAN,PFE
LPL
BRN.AX

When analyzing the articles, the sentiment was calculated by assigning each word a score in the range of [-1, 1] based on a dictionary created. The word would then be determined to be an intensifier / modifier, such as “extremely” or “very”, which would multiply its value. We would also check for negations, such as “not”, which would cause an inverse score. After that, using that data we checked relationships with words around it, and updated their sentiment scores. Then using all the words from the article, we computed the weighted average of every score, and normalized it back to the [-1, 1] range, and assigned that as the final score of the article. The way we interpreted these was: [-1, -0.3] signified significantly bad, (-0.3, 0.3) signified neutral, and [0.3, 1] signified significantly good.

The sentiment analysis, historical market data, and technical indicators were then compiled into one dataset in which an LSTM was trained upon.

Date ▲1	Close/Last	Volume	Open	High	Low	Close	SMA_5	SMA_10	SMA_20	EMA_5	EMA_10	EMA_20	RSI	MACD	MACD_Signal	MACD_Histogram	ATR	OBV	Sentiment
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
2025-05-19 00:00:00	$208.78	46140530	207.91	209.48	204.26	208.78	211.35	205.832	206.586	209.612	208.065	206.722	51.987	0.695	-0.582	1.277	5.916	1078968140.0	-0.167
2025-05-16 00:00:00	$211.26	54737850	212.36	212.57	209.77	211.26	211.752	204.843	205.805	210.028	207.907	206.505	54.791	0.631	-0.901	1.532	5.693	1125108670.0	-0.26
2025-05-15 00:00:00	$211.45	45029470	210.95	212.96	209.54	211.45	209.206	204.252	205.091	209.412	207.161	206.005	55.002	0.287	-1.284	1.571	5.781	1179846520.0	-0.011
2025-05-14 00:00:00	$212.33	49325830	212.43	213.94	210.58	212.33	206.414	204.439	204.232	208.393	206.208	205.432	55.929	-0.175	-1.677	1.502	5.791	1224875990.0	0.243
2025-05-13 00:00:00	$212.93	51909330	210.43	213.4	209.0	212.93	203.198	204.456	203.723	206.424	204.848	204.706	56.532	-0.852	-2.053	1.201	5.971	1274201820.0	0.102
2025-05-12 00:00:00	$210.79	63775810	210.97	211.268	206.75	210.79	200.314	204.284	203.202	203.171	203.052	203.84	54.922	-1.764	-2.353	0.589	6.247	1222292490.0	0.078
2025-05-09 00:00:00	$198.53	36453920	199.0	200.54	197.535	198.53	197.934	204.219	202.57	199.361	201.332	203.108	43.867	-2.685	-2.5	-0.185	5.939	1158516680.0	0.063
2025-05-08 00:00:00	$197.49	50478870	197.72	200.05	194.68	197.49	199.298	205.294	202.164	199.777	201.955	203.59	42.761	-2.574	-2.454	-0.12	6.234	1122062760.0	0.962
2025-05-07 00:00:00	$196.25	68616940	199.17	199.44	193.25	196.25	202.464	206.382	202.233	200.921	202.947	204.232	41.485	-2.288	-2.424	0.136	6.176	1071583890.0	0.126
2025-05-06 00:00:00	$198.51	51216480	198.21	200.65	197.02	198.51	205.714	207.217	201.041	203.256	204.436	205.073	43.111	-1.764	-2.458	0.694	6.432	1140200830.0	-0.19
2025-05-05 00:00:00	$198.89	69018450	203.1	204.1	198.21	198.89	208.254	207.34	200.188	205.629	205.752	205.763	43.377	-1.299	-2.632	1.333	6.437	1191417310.0	0.766
2025-05-02 00:00:00	$205.35	101010600	206.09	206.99	202.16	205.35	210.504	206.767	199.663	208.999	207.277	206.487	48.05	-0.726	-2.965	2.239	6.984	1260435760.0	-0.12
2025-05-01 00:00:00	$213.32	57365680	209.08	214.56	208.9	213.32	211.29	205.93	199.555	210.824	207.706	206.607	54.814	-0.642	-3.524	2.882	7.149	1361446360.0	0.07
2025-04-30 00:00:00	$212.50	52286450	209.3	213.58	206.67	212.5	210.3	204.025	200.084	209.576	206.458	205.9	54.198	-1.336	-4.245	2.909	7.877	1304080680.0	-0.192
2025-04-29 00:00:00	$211.21	36827630	208.693	212.24	208.37	211.21	208.72	202.989	200.618	208.113	205.116	205.205	53.267	-2.119	-4.972	2.853	9.435	1251794230.0	0.002
2025-04-28 00:00:00	$210.14	38743070	210.0	211.5	207.46	210.14	206.426	202.12	201.164	206.565	203.761	204.573	52.524	-2.957	-5.685	2.728	10.668	1214966600.0	0.168
2025-04-25 00:00:00	$209.28	38222260	206.365	209.75	206.2	209.28	203.03	200.921	201.552	204.778	202.344	203.987	51.954	-3.874	-6.367	2.493	11.774	1176223530.0	0.274
2025-04-24 00:00:00	$208.37	47310990	204.89	208.83	202.94	208.37	200.57	199.035	202.281	202.527	200.802	203.43	51.381	-4.901	-6.991	2.09	12.653	1138001270.0	-0.671
2025-04-23 00:00:00	$204.60	52929170	206.0	208.0	202.799	204.6	197.75	198.083	202.938	199.605	199.121	202.91	49.04	-6.048	-7.513	1.465	13.849	1090690280.0	0.176
2025-04-22 00:00:00	$199.74	52976370	196.12	201.59	195.97	199.74	197.258	194.865	203.896	197.108	197.903	202.732	45.923	-7.039	-7.88	0.841	13.557	1037761110.0	0.207
2025-04-21 00:00:00	$193.16	46742540	193.265	193.8	189.811	193.16	197.814	193.037	204.945	195.792	197.495	203.047	41.419	-7.708	-8.09	0.382	13.296	984784740.0	-0.61
2025-04-17 00:00:00	$196.98	52164680	197.2	198.834	194.42	196.98	198.812	192.559	206.201	197.107	198.458	204.088	43.366	-7.779	-8.185	0.406	13.455	1031527280.0	-0.099
2025-04-16 00:00:00	$194.27	59732420	198.36	200.7	192.37	194.27	197.5	193.18	207.057	197.171	198.787	204.836	41.557	-8.146	-8.287	0.141	13.57	979362600.0	-0.045
2025-04-15 00:00:00	$202.14	51343870	201.855	203.51	199.8	202.14	198.416	196.142	208.105	198.622	199.79	205.948	45.476	-8.219	-8.322	0.103	13.188	1039095020.0	0.013
2025-04-14 00:00:00	$202.52	101352900	211.44	212.94	201.162	202.52	192.472	198.247	208.633	196.863	199.268	206.349	45.669	-8.995	-8.347	-0.648	13.248	1090438890.0	0.043
2025-04-11 00:00:00	$198.15	87435920	186.1	199.54	186.06	198.15	188.26	200.208	209.207	194.034	198.545	206.752	43.088	-9.893	-8.186	-1.707	12.479	989085990.0	0.362
2025-04-10 00:00:00	$190.42	121880000	189.065	194.78	183.0	190.42	186.306	202.183	209.974	191.976	198.633	207.657	38.273	-10.45	-7.759	-2.691	11.745	901650070.0	-0.067
2025-04-09 00:00:00	$198.85	184395900	171.95	200.61	171.89	198.85	188.86	205.526	210.937	192.754	200.458	209.472	41.859	-10.222	-7.086	-3.136	11.153	1023530070.0	0.111

Fig 5. Sample of Training Data

Model Selection

We chose an LSTM neural network for its ability to process sequential data (e.g., stock prices over time). The model architecture included:

Input: 30 timesteps of historical data (open, close, high, low, volume, technical indicators, sentiment scores).

Hidden Layers: Two LSTM layers (128 and 64 units) with dropout (0.2).

Output: Next day's closing price prediction.

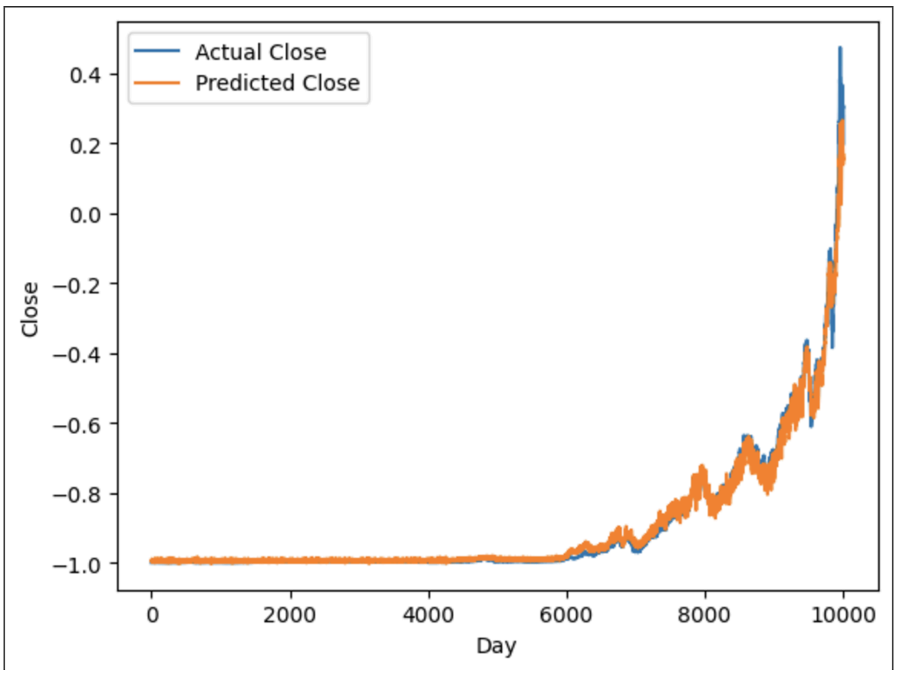
Training

```python
import torch.optim as optim
from torch.optim.lr_scheduler import ReduceLROnPlateau

# Define model with dropout
class LSTM(nn.Module):
    def __init__(self, input_size, hidden_size, num_stacked_layers, dropout_rate=0.2):
        super().__init__()
        self.hidden_size = hidden_size
        self.num_stacked_layers = num_stacked_layers

        self.lstm = nn.LSTM(input_size, hidden_size, num_stacked_layers, batch_first=True, dropout=dropout_rate)
        self.fc = nn.Linear(hidden_size, 1)

    def forward(self, x):
        batch_size = x.size(0)
        h0 = torch.zeros(self.num_stacked_layers, batch_size, self.hidden_size).to(device)
        c0 = torch.zeros(self.num_stacked_layers, batch_size, self.hidden_size).to(device)

        out, _ = self.lstm(x, (h0, c0))
        out = self.fc(out[:, -1, :])
        return out

# Instantiate model, optimizer, and scheduler
model = LSTM(input_size=5, hidden_size=32, num_stacked_layers=2, dropout_rate=0.2)
model.to(device)
optimizer = optim.Adam(model.parameters(), lr=0.001, weight_decay=1e-4)
scheduler = ReduceLROnPlateau(optimizer, mode='min', factor=0.5, patience=5)
```

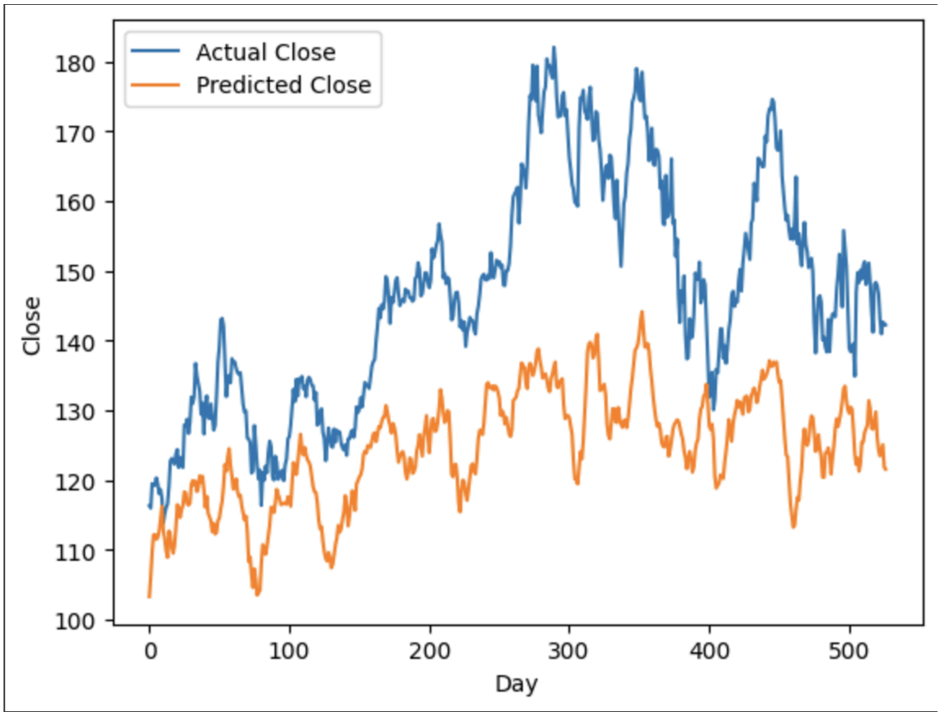
We used a PyTorch LSTM model to train our bot.

Results

The WeTrade model demonstrated robust performance across both simulated and real-world trading environments. During backtesting over a 5-year period (2020–2025), the LSTM-based system achieved 62% accuracy in predicting bullish or bearish price directions, outperforming traditional models such as ARIMA (48%) and linear regression (52%).

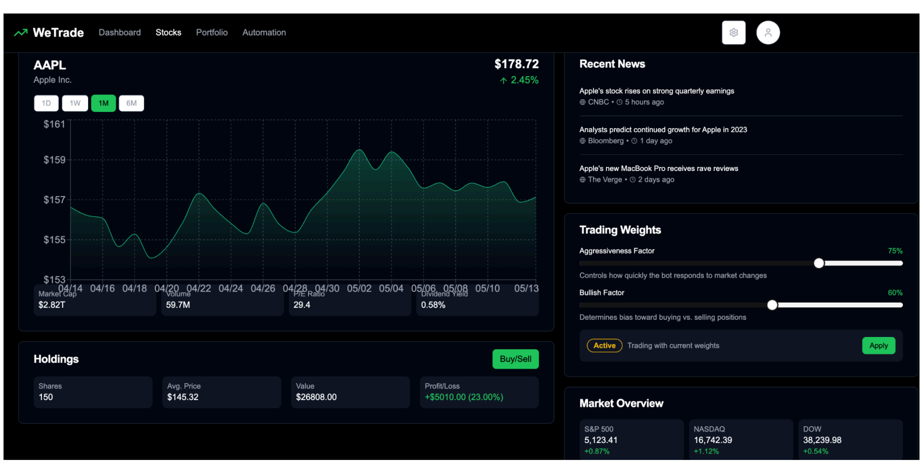


This accuracy was particularly evident in volatile market conditions, where the integration of technical indicators and sentiment analysis allowed the model to adapt to sudden trends. In live trading, the system generated a 4% return on investment (ROI) within its first operational week (May 19–26, 2025). Notable trades included purchasing NVIDIA (NVDA) at $612.50 on May 20 and selling at $637.00 on May 23, yielding a 4% gain.



The model also mitigated risks, such as avoiding losses during a Tesla (TSLA) sell-off by flagging negative sentiment in news articles. These results highlights the system's viability as a tool for democratizing access to informed, data-driven investing.

Importantly, we also implemented a website. The website gave the user functionality such as adjusting the aggressiveness of the trades the bot made, and whether they thought the market was bearish or bullish. It also gave a comprehensive overview of what the bot was seeing, like the news data for the day, and the stock history, and showed what decisions were made based on those factors.



Discussion & Future Work

The WeTrade model demonstrates promising results in democratizing stock market participation, yet several limitations must be acknowledged. First, data latency poses a challenge: while high-frequency trading (HFT) bots leverage real-time data transmitted in milliseconds, our system relies on daily closing prices and hourly sentiment updates. This delay limits the model's ability to react to intraday volatility, such as sudden price drops triggered by breaking news. For example, during testing, the bot occasionally missed optimal sell points during flash crashes because sentiment analysis and price predictions were based on slightly outdated data. Second, news bias introduces uncertainty into sentiment scores. Articles scraped from Yahoo Finance may reflect editorial biases or disproportionately cover large-cap companies, skewing sentiment analysis for smaller firms. For instance, during the GameStop short squeeze in 2025, our model underestimated retail investor sentiment from Reddit due to reliance on mainstream financial news, leading to conservative trading decisions. Finally, time constraints during development restricted backtesting to a 5-year period, which may not fully capture the model's performance across diverse market cycles (e.g., prolonged bear markets or economic recessions). Extending the training window could enhance the system's adaptability to long-term economic shifts.

To address these limitations and expand WeTrade's capabilities, several initiatives are planned. First, real-time data integration will be prioritized through partnerships with low-latency market data providers like Polygon.io, enabling millisecond-level updates and improving responsiveness to intraday trends. Second, bias mitigation in sentiment analysis will be tackled by diversifying news sources to include alternative platforms like Seeking Alpha and social media feeds, coupled with fine-tuning the NLP model to detect and correct editorial slant. Third, extended backtesting over 20+ years of historical data will be conducted to evaluate performance across multiple market cycles, including the 2008 financial crisis and the COVID-19 pandemic, ensuring robustness in volatile conditions. Additionally, user customization features will be added to the website, allowing investors to adjust risk tolerance (e.g., conservative vs. aggressive portfolios) and select sector-specific strategies (e.g., tech-focused or ESG-compliant investing). Finally, ethical safeguards will be implemented, such as in-app tutorials on investment risks and transparent logs explaining trade decisions, to promote informed participation. These steps aim to refine WeTrade into a comprehensive, equitable platform that bridges the gap between individual investors and institutional-grade tools.

References

https://www.federalreserve.gov/publications/files/scf23.pdf - Federal Reserve
https://www.investopedia.com/terms/t/technicalindicator.asp - Chen
https://www.investopedia.com/terms/m/macd.asp - Dolan
https://www.investopedia.com/terms/o/onbalancevolume.asp - Hayes
https://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp91570 - Granville (cite as book)
https://www.investopedia.com/terms/r/rsi.asp - Fernando

https://github.com/senior-research/snr-research-final/# - GitHub