Final project Midterm Check-in:

1. Objective

This project aims to design a deep-learning-based framework for portfolio optimization and predictive trading. The project aims to predict buy/sell signals and optimal entry prices for stocks using a hybrid combination of LSTM (for temporal forecasting) and CNN + SVM (for pattern-based classification).

The study uses the S&P 500 as a reference universe specifically for the top 7 and bottom 7 performers with VOO (Vanguard S&P 500 ETF) serving as a benchmark for evaluation.

The aim is to incorporate (via Polygon.io) once able to show results daily; current experiments use daily OHLCV data (2017–2025) from Yahoo Finance.

The goal is twofold:

Originally, these were in two stages. First is to develop a Buy/sell sign followed by Price Prediction. After some testing and thoughts. I decided to flip it around. Stage 1 will be price predicted, followed by buy/sell sign.

Price Prediction – Use machine learning models to predict precise entry prices, thereby improving portfolio performance relative to the S&P 500 benchmark.

Buy/Sell Signals – Identify optimal entry and exit points by applying deep learning methods to stock data transformed into images.

2. Literature Review

Early work in financial prediction has explored a range of methods, from technical analysis and econometrics to deep neural architectures.
Key sources reviewed include:

B, Heaton J, et al. “Deep Learning in Finance.” ArXiv.org, 2016, arxiv.org/abs/1602.06561.

Cohen, Naftali, et al. “Trading via Image Classification.” ArXiv.org, 2019, arxiv.org/abs/1907.10046. Accessed 15 Sept. 2025.

Feng, Guanhao, et al. “Deep Learning for Predicting Asset Returns.” ArXiv.org, 26 Apr. 2018, arxiv.org/abs/1804.09314.

Zhang, Cheng, et al. Deep Learning Techniques for Financial Time Series Forecasting: A Review of Recent Advancements: 2020-2022. 20 Apr. 2023, https://doi.org/10.48550/arxiv.2305.04811.

Cohen, Balch & Veloso (2019) – *Trading via Image Classification*
Introduced the concept of converting time-series price data into candlestick images and using CNNs for trading signal prediction.

Zhang et al. (2017) – *Stock Market Prediction via LSTM Networks*
Demonstrated how recurrent networks capture temporal dependencies in financial time series.

Selvin et al. (2017) – *Stock Price Prediction Using LSTM, CNN, and RNN*
Compared hybrid architectures and found CNN-LSTM combinations to perform best for short-term forecasting.

Bao, Yue, & Rao (2017) – *A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders*
Illustrated unsupervised pretraining for robust representation of noisy market data.

Zhao et al. (2023) – *Hybrid CNN-SVM for Stock Price Movement Prediction*
Proposed replacing the Softmax output layer in CNNs with an SVM classifier, improving boundary sharpness and generalization.

Identified Gap:
Most prior studies use either CNNs or LSTMs in isolation, or train end-to-end classification pipelines without modular decision layers. Few integrate sequential forecasting (via LSTM) with pattern-based classification (via CNN+SVM).
This project targets that gap by:

- Using LSTM to predict *next-day prices (Stage 1)*, and
- Feeding these forecasts into a CNN-SVM decision layer (Stage 2) to generate actionable trading signals.

3. Dataset Preparation and EDA

Data Source

- VOO (Vanguard S&P 500 ETF) and others are via Yahoo Finance (yfinance API).

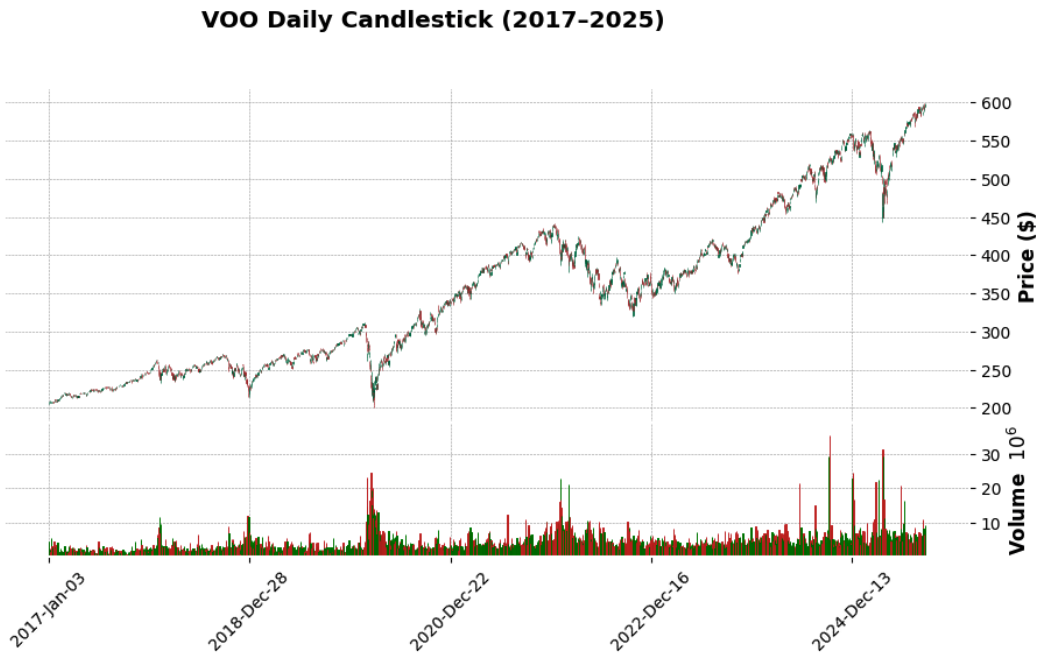
- Period: Jan 2017 – Sept 2025: This is because I am applying for LSTM/SVM.
- Features: Open, High, Low, Close, Volume.

Cleaning and Preparation

- Removed missing entries and non-trading days. Load in only trading day
  - Select data 2017-01-01 to 2025-09-10
- Select only OHLC-V
  - Here is an example

Date	Open	High	Low	Close	Volume
2017-01-03 00:00:00-05:00	206.679993	207.330002	205.559998	206.740005	4750200
2017-01-04 00:00:00-05:00	207.199997	208.179993	207.119995	207.960007	4622400
2017-01-05 00:00:00-05:00	207.750000	208.039993	207.009995	207.800003	2772100
2017-01-06 00:00:00-05:00	207.990005	209.089996	207.399994	208.610001	2194600
2017-01-09 00:00:00-05:00	208.339996	208.479996	207.889999	207.949997	1705200

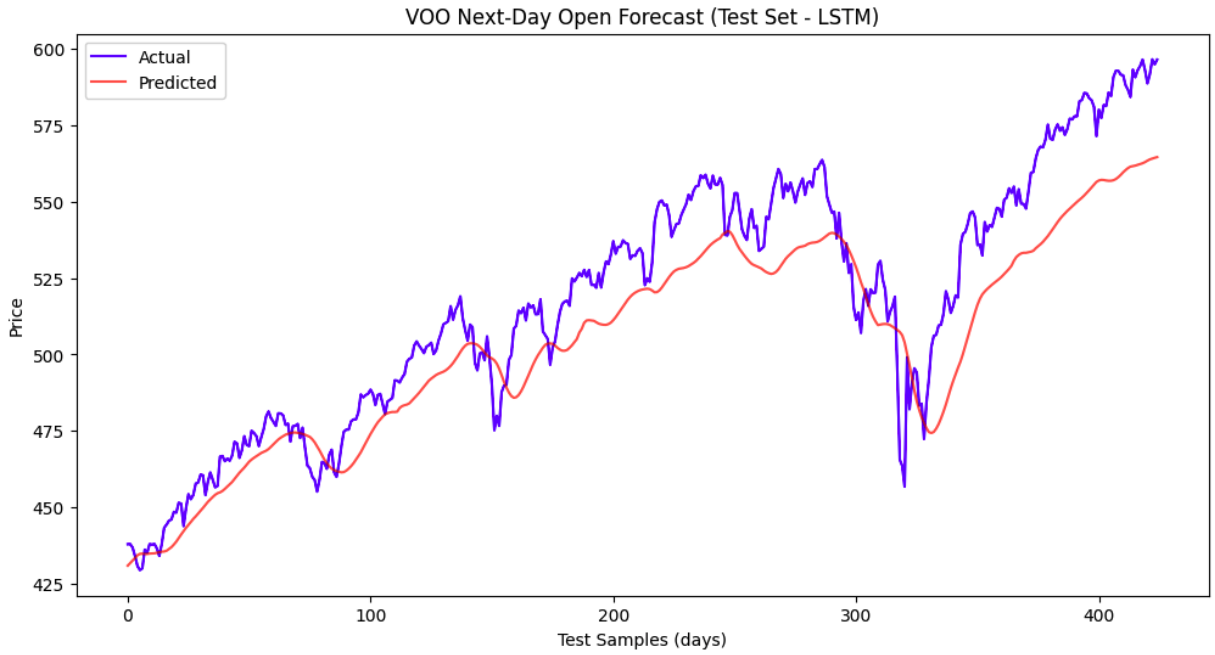
From OHLC-V, I converted into a time series candlestick as below.



The data will be split in train and test (80/20)

Stage 1:

- Apply CNN vs Apply LSTM, the initial ideas to build a signal, but after many testing. The application does not seem to be the best solution. Leading to switching into prediction. Below is example of the price prediction using LSTM. We see that it is tracing well with trends and seasonality.



EDA Highlights

- LSTM – we can do 10,20,30,45, 60 days rowing and faster compare to CNN.

Visual Analysis

- Candlestick plots and Bollinger bands reveal clustering of volatility during market downturns (e.g., COVID-19 crash, 2022 inflation period).
- The LSTM later used x-day lookbacks as baseline, refined through cross-validation.

4. Problem Definition

Stage 1 - Research Question:

How can hybrid deep learning architectures (LSTM for sequential forecasting, CNN+SVM for decision boundary classification) improve short-term predictive accuracy and trading efficiency compared to traditional or single-model methods?

Input (SVM or LSTM):

- Past *N*-day sequences of OHLCV data (10–60 days).

Output:

- Stage 1: Predicted next-day price (regression).
- Stage 2: Buy/Sell/Hold classification signal (binary or ternary).

Evaluation Metrics:

- Stage 1 (forecasting): Mean Squared Error (MSE), Mean Absolute Error (MAE), and correlation between predicted vs. actual price.
- Stage 2 (classification): Accuracy, Precision/Recall, and financial metrics like Sharpe ratio during back testing.

5. Model Plan

Stage 1 – LSTM/SVM Forecasting

- Architecture: Two LSTM layers (64 units each) + Dropout (0.2) + Dense(32→1).
- Input: 10–60 day rolling OHLCV sequences.
- Output: Next-day Close price (scaled).
- Optimization: Early stopping on validation loss; tuned lookback window (10–60 days).

Stage 2 – CNN+SVM Decision Model (planned)

- Stage 2 Input:
    - Candlestick images (visual patterns of past 20 days).
    - LSTM-predicted price or return from Stage 1.

- Model:
  - CNN as a *feature extractor*.
  - Final layer replaced with SVM instead of Softmax for sharper decision margins.
- Goal:
  - Classify whether the next-day move exceeds ±0.5% threshold → Buy / Sell / Hold.
  - Avoid overfitting by separating pattern learning (CNN) from decision logic (SVM).

Baseline Comparisons

- Bollinger Band rule-based labels – served as the initial labeling benchmark.
- SVM with tabular OHLCV features – achieved ~53–56% directional accuracy.
- CNN regression on candlestick images – underfit (predicted near-zero returns), motivating the hybrid redesign.

6. Progress Summary

Stage	Description	Status
Literature Review	Expanded to 7+ academic papers from	Done
Dataset Prep & EDA	Cleaned, normalized, exploratory plots – ensure proper data pipeline	Done
	CNN on candlestick images (underfit)	Done
	Switched to returns instead of raw price	Done
Alternative Models	SVM on OHLCV statistical features	IP
Stage 1	LSTM forecasting with window tuning	IP
Stage 2	CNN+SVM hybrid decision layer	IP

7. Next Steps

1. Finalize Stage 2 (CNN feature extractor + SVM classifier).
2. Integrate LSTM-predicted prices as additional inputs for CNN-SVM.

3. Perform back testing of generated buy/sell signals against a Buy-and-Hold benchmark.
4. Compute Sharpe ratio, drawdown, and cumulative return metrics.